ANO 1 - NÚMERO 8
CR$

interface

HARDWARE · SOFTWARE · COMPUTADORES · MINI · MICRO · TELEPROCESSAMENTO

FILIA:
UM SISTEMA DE PROGRAMAÇÃO

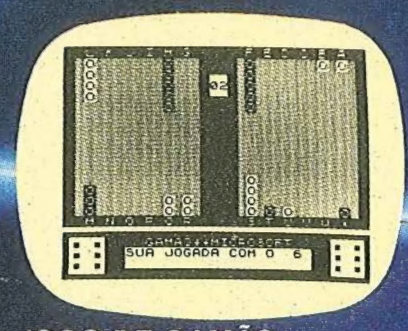
memórias

MUMPS

CONCURSO
TK 82-C

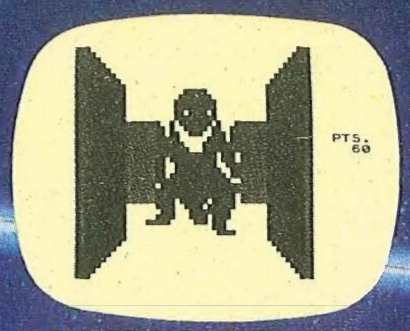
MICROSOFT

Programas para o seu TK82-C e TK85



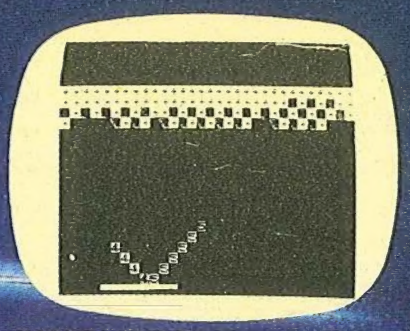
JOGO DE GAMÃO
16K

Este programa apresenta o tabuleiro no video e utiliza o eficiente código de máquina, permitindo 4 (quatro) níveis de dificuldades de jogo.



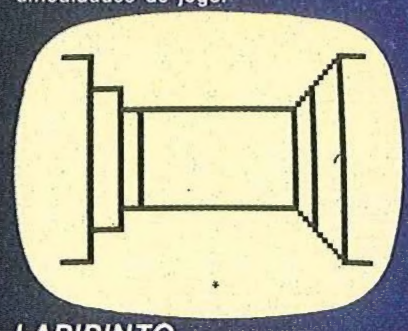
MONSTRO DAS TREVAS TRIDIMENSIONAL · 16K

Impressionante jogo onde você deve evitar o monstro das trevas. Tudo em 3 dimensões.



DEMOLIDOR
2K

Jogo animado, tipo "fliperama". O jogador deverá demolir uma parede, com uma bola que se encontra sempre em movimento.



LABIRINTO TRIDIMENSIONAL · 16K

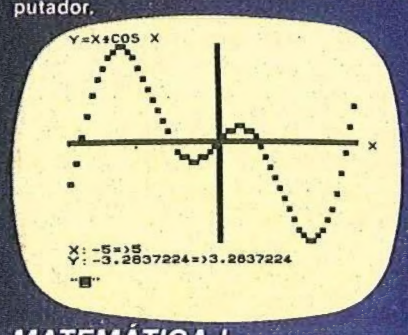
Jogo em três dimensões. O jogador poderá definir a dificuldade do Labirinto. O programa apresenta a posição do jogador em perspectiva. Em qualquer momento é possível pedir auxílio ao computador.

INVASORES DO ESPAÇO · 16K

Consiste de uma frota de naves invasoras extraterrenas, descendo no planeta Terra. Sua missão é destruir as naves invasoras dispondo da arma de raios laser.

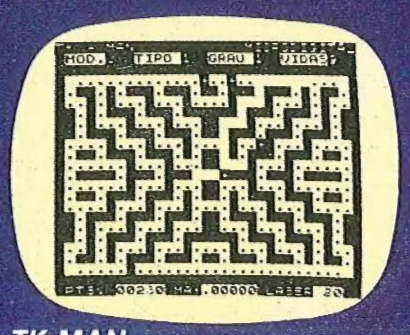
RALLY
16K

Emocionante corrida de rally em um labirinto, onde poderão ser testados sua habilidade e seus reflexos. Para conseguir seu intento, você deverá evitar carros-ataque e obstáculos em seu trajeto.



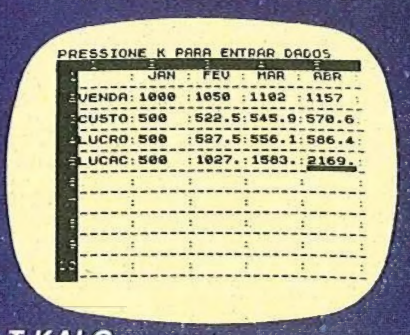
MATEMÁTICA I
16K/64K

Análise gráfica de funções matemáticas, resolução de sistemas de equações lineares (16K-51 equações/64K-95 equações), e Cálculo de integrais definidos.



TK-MAN
16K

Jogo animado onde deverão ser apagados todos os pontinhos espalhados em um labirinto (o programa contém 15 tipos de labirinto). Você será impedido a qualquer custo, por 4 extraterrenos, guardiães do labirinto, que poderão ser combatidos com cargas de raios-laser.



T-KALC
16K/64K

Programa desenvolvido para cálculos numéricos em planilha. O usuário define as colunas, as linhas e as fórmulas aplicadas. Similar ao famoso Visicalc. De grande versatilidade, este programa permite a formulação de cálculos científicos e comerciais, análise de tabelas numéricas e outras aplicações.

TKADREZ II
16K

Este jogo apresenta o tabuleiro e as peças no vídeo. Permite a escolha de até 7 níveis de dificuldade. O programa fornece a qualquer momento, a listagem dos lances efetuados, e armazena em fita a posição das peças. Ele poderá recomendar a sua jogada.

NOS REVENDEDORES AUTORIZADOS EM TODO PAÍS

80 programas à sua disposição. Solicite folheto.

MICROSOFT®

Rua do Bosque, 1.234 - PABX 825-3355
CEP 01136 - Cx. Postal 54.121 - São Paulo-SP

ANTENNA - EDIÇÕES TÉCNICAS LTDA.
FILIAL DE SÃO PAULO
RUA VITÓRIA, 379/383

conteúdo

interface

matérias

Filia: Um Sistema de Programação para Microcomputadores (1ª Parte) 12

Filia comparada às linguagens tradicionais. Os primeiros passos em programação com Filia.

Por Rildo Pragana

Microprocessador BIT SLICED: Uma Opção Para 32 Bits (4ª Parte) 16

Circuitos seqüenciadores LSI

O Sistema Operacional CP/M (5ª Parte) 28

Funções do BDOS relativas a disco

Por Valdelírio P. S. Filho (Microsol Ind. Com. e Serv. Eletrônicos Ltda)

Curso Microprocessador Z-80 (7ª Lição) 31

Software do Z-80: aritmética de complemento a dois (2's), bits de condição com operações lógicas e aritméticas e sintaxe da linguagem ASSEMBLY do Z-80.

Por André Gil Rubens e Ney Acyr de Oliveira (A & N – Consultoria em Hardware e Software de Microcomputadores)

Memórias (1ª Parte) 42

Histórico dos dispositivos de armazenamento, tipos e características.

Por Cesar da Costa

Mumps: Características e Comparações Com Outros Sistemas de Programação (1ª Parte) 46

Por T. Munneke; R. F. Walters; J. Bowie; C. B. Lazarus e D. A. Bridger (Tradução MEDIDATA-RJ)

Mercado de Computadores 51

Dicas (Programação de ROM)

Pólo de Informática

departamentos

Notas do Editor 02
Informe 04
Consultoria 06
Hardhouse 08
Calendário 22
Livros 41
Novos Produtos 45
Cartas 50
Páginas de Serviço 59

seções

Controladores/Interfaces (2ª Parte) 23

Projeto de separadores de dado e clock para acionadores de discos flexíveis

Por Manuel Lois Anido

Clube do Microsoftware 37

Microcomputadores (2ª Parte) 38

Arquivo de dados em disco versus arquivo de dados em cassete

Por Cesar da Costa

PÓLO DE INFORMÁTICA

DIRETORES RESPONSÁVEIS:
Naila Glória O. Côrtes
Cesar da Costa
EDITOR:
José Levy Fidelix da Cruz
EDITORES TÉCNICOS:
Naila Glória O. Côrtes
Cesar da Costa
CONSULTORIA TÉCNICA:
Tarcisio Neves da Cunha
Manuel Lois Anido
José Arthur da Rocha
André Gil Rubens
Ney Acyr de Oliveira
Cesar de Araújo Lima
Valdelírio P. S. Filho
José Vicente M. Borges
PRODUÇÃO/DIAGRAMAÇÃO:
Nilton Luiz Côrtes
ARTE FINAL:
Jorge Gatto
REDAÇÃO/REVISÃO:
Nelma Bernadete C. de Castro
DIREÇÃO COMERCIAL/PUBLICIDADE:
Success — Assessoria, Planejamento e Marketing Ltda
DIRETOR COMERCIAL:
José Levy Fidelix da Cruz
Contatos — Rio de Janeiro:
José Augusto N. Rodrigues
Milton Alves Junior
Roberto Passeri
Endereço:
Rua Senador Dantas, 117/Sala 1139
Tel.: (021) 240-5200
São Paulo:
Success A. P. M. Ltda
Tel.: 34-3305 e 34-7255
Kapron Propaganda Ltda
Tel.: 223-2037
Contatos — Rio Grande do Sul:
Francisco Linck
Rua Osório Ramos Correa, 220
Tel.: (0512) 88-1252

REPRESENTANTES (VENDA/ASSINATURAS):
Up-date — Publicações e Serviços Ltda (SP)
Tel.: (011) 813-1375
Kristian Eletrônica Ltda (RJ)
Tel.: (021) 262-7119
Ítalo Humberto Frigério (RJ)
Tel.: (021) 220-2109
Spartime — Serviços de Publicidade (RS)
Tel.: (0512) 24-2955
Oficcina Minis e Microcomputadores Ltda (BA/PE)
Tel.: (071) 248-6666 R/268 (081) 326-9318
Compushow Computadores Ltda (DF)
Tel.: (061) 273-2128
Microsol — Software, Comércio e Manutenção Ltda (CE)
Tel.: (085) 227-5878
Altair Almeida Rios (RJ)
Caixa Postal 36021 — Rio de Janeiro — CEP 20712
Sagittarius — Comércio e Representação Ltda
Tel.: (021) 234-3599
Assinaturas para o Brasil: Cr$ 5.000,00 e para o exterior US$ 50,00 (1 ano)
Administração:
Alcenir Fernandes

TIRAGEM:
20.000 exemplares
DISTRIBUIÇÃO:
Publicações Castro Ltda
Tel.: (021) 262-1226 e (011) 228-5076

COMPOSIÇÃO:
Damião Moraes

FOTOLITOS E IMPRESSÃO:
Gráfica Editora Lord S.A.
Interface é editada pela Prodigt — Processamento, Tecnologia e Comunicação Ltda.
Administração e Redação:
Estrada do Tindiba, 2380 — Jacarepaguá — Rio de Janeiro — RJ — CEP 22700 — Tel.: 392-8965
Interface não aceita matéria redacional paga. Todos os direitos de reprodução desta publicação estão reservados, só podendo ser reproduzidos com fins comerciais, mediante autorização prévia.
Os artigos assinados são de responsabilidade do autor, as opiniões emitidas não são necessariamente coincidentes com as da Revista.
Registro no INPI protocolo nº 810965755.

notas do editor

"A crítica talvez mais comum à Universidade brasileira é a relativa ao seu alheamento dos problemas mais prementes das comunidades que as abrigam.

Considero-a crítica injusta. Muitas vezes, é a sociedade, são os governos, as empresas que não dirigem suas solicitações à Universidade, criando mecanismos paralelos de formação técnico-profissional e de pesquisa aplicada, ao invés de procurarem mobilizar o potencial existente nas instituições de ensino e pesquisa. Ou é a Universidade que produz respostas inadequadas a essas demandas sociais.

É importante, no entanto, que as Universidades procurem orientar-se mais "para fora", com vistas a uma mais produtiva inserção de suas respostas a esses reclamos sociais. E que tomem a iniciativa de provocar importantes e dinâmicas interações com a sociedade" (Roberto Cavalcanti de Albuquerque em **A Universidade e o Nordeste**).

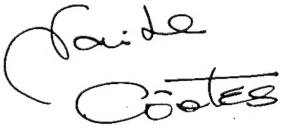
A orientação mais "para fora" das Universidades brasileiras calará de vez os tecnocratas que por outros interesses, escusos a estas, acusam a tecnologia nacional de obsoleta.

INTERFACE através da seção EDUCAÇÃO/TECNOLOGIA abre espaço para as Universidades brasileiras, espaço para divulgarem seus trabalhos, para mostrarem à sociedade seu verdadeiro momento. Também àqueles que queiram divulgar seus trabalhos de final de curso ou teses de mestrado, estamos "abertos".

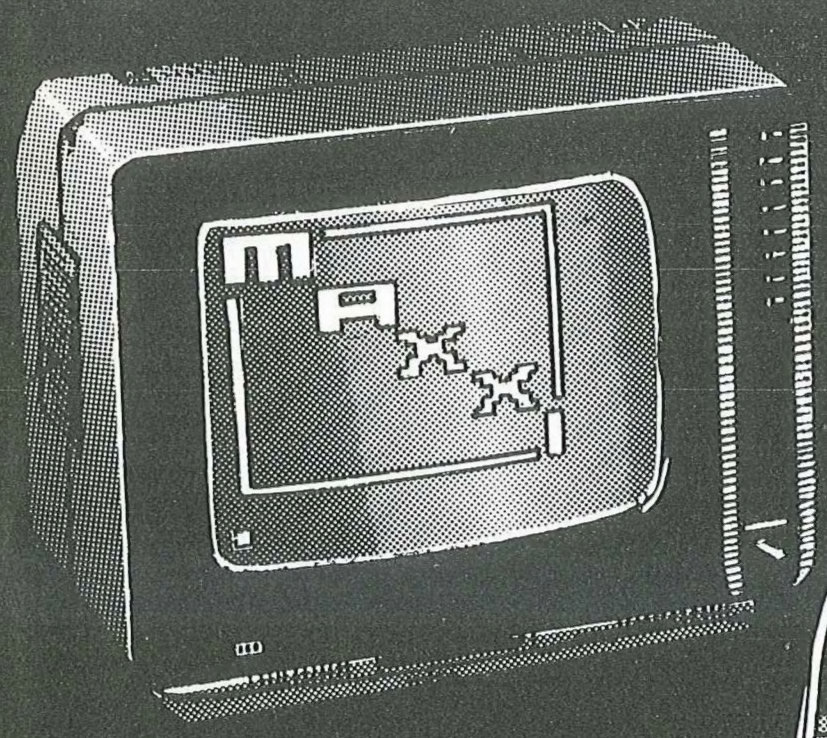
E iniciando este processo com as Universidades, publicamos o trabalho do Prof. Rildo Pragana, do Departamento de Física da Universidade de Pernambuco, FILIA: UM SISTEMA DE PROGRAMAÇÃO PARA MICROCOMPUTADORES. Linguagem desenvolvida por ele e sua equipe, teve como objetivo inicial o controle de processos, já estando no entanto implementada para diversos microcomputadores pessoais.

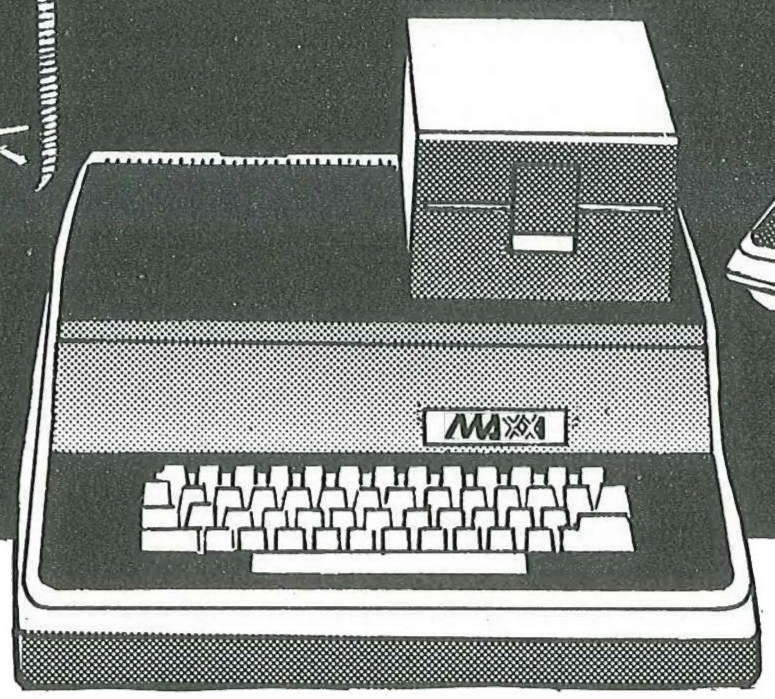
O CURSO MICROPROCESSADOR Z-80 está entrando no software nesta edição, segmento tão aguardado pelos programadores em geral.

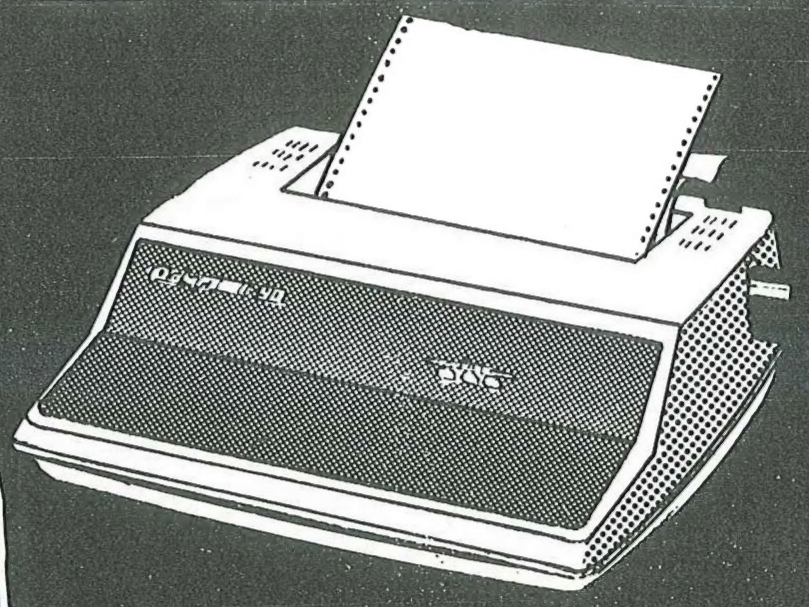
NÃO FIQUE SÓ NA TEORIA — INTERFACE lhe reserva uma boa surpresa. Aguarde.



O MICRO CHEGOU AO MAXXI.







E O MENOR PREÇO JÁ ESTÁ COMPUTADO NA CLAPPY.

O Maxxi chegou para minimizar o seu trabalho e facilitar a sua vida: administra sua conta bancária, elabora sua declaração de renda, organiza seu orçamento doméstico e até auxilia o estudo de seus filhos.
Nas áreas científicas e educacionais, o Maxxi também faz escola. É especialmente indicado para operações em colégios, universidades, centros de pesquisas, centrais de informação, cursos de formação, treinamento e inúmeras outras áreas do setor.
O Maxxi é também um equipamento indispensável ao moderno empresário ou profissional liberal: põe em dia a administração da sua empresa (contabilidade, controle de estoque, contas a receber), o seu planejamento financeiro e inúmeros outros assuntos profissionais. Venha a Clappy conhecer o Maxxi por um preço micro.

GRÁTIS.
Programa para planejamento financeiro e processador de texto.

CARACTERÍSTICAS PRINCIPAIS

Compatibilidade total com APPLE II PLUS. Conexão com vídeo, monitor ou televisor comercial (colorido ou preto e branco) operando no Modo Texto, em vídeo normal, reverso ou piscante com 40 caracteres/linha, 24 linhas. No Modo Gráfico, em baixa resolução, com 16 cores selecionáveis por SOFTWARE. Em alta resolução, com 6 cores selecionáveis por SOFTWARE. Memória de 48 K. Interface para gravador cassete, vídeo, um conjunto de 8 conectores para a ligação de cartões controladores de periféricos.

PERIFÉRICOS

TV comercial, colorido ou preto e branco. Monitor profissional Polymax, com fosfatização verde. Unidades de disquete de 5 1/4'', sendo que cada disquete é composto de 35 trilhas, 16 setores e 256 bytes. Gravador cassete (comercial) utilizado para armazenar programas, tabelas, gráficos. Impressora Polyprint 90, com velocidade de 90 CPS. Controles de jogos. Interface serial para comunicação de dados. MODEM. Interface serial para impressora. Cartão 80 colunas (VIDEX). Cartão de expansão de 16 K. Cartão CP/M.

BREVE **Clappy** COPACABANA
Rua Pompeu Loureiro, 99

Polymax
SISTEMAS E PERIFÉRICOS S.A

Clappy

Venha à nossa loja ou solicite a visita de um representante.

Centro: Av. Rio Branco, 12 - loja e sobreloja
Centro: Rua Sete de Setembro, 88 - loja O (galeria)
São Cristóvão: Rua Antunes Maciel, 25 - 2.º andar
253-3170 • 253-3395 • 283-3588
228-0734 • 248-8159 •

Entregamos em todo o Brasil pelo reembolso VARIG.

RESERVA DE MERCADO

COMUNICADO APPD-RJ

Os profissionais de Processamento de Dados do Rio de Janeiro, através de sua Associação, APPD-RJ, vêm se manifestando ao longo do tempo, a favor da Política de Reserva de Mercado para Minis e Microcomputadores.

Consideramos que o mecanismo de Reserva de Mercado foi o principal fator para o sucesso da criação e consolidação da Indústria Nacional de Informática. Existem hoje mais de 40 empresas brasileiras atuando no setor, gerando empregos para mais de 8000 trabalhadores criando tecnologia local e participando com mais de 42% do mercado de computadores em apenas 6 anos de existência. Em particular, na área de mini e microcomputadores, a participação da indústria nacional chega a 80% do mercado.

A capacitação tecnológica alcançada a partir da implementação da Política de Reserva de Mercado é um Patrimônio Nacional, do qual o país não pode abrir mão, apenas para atender aos anseios de domínio do mercado brasileiro de computação.

Observa-se, no momento atual, ameaças à política de Reserva do Mercado, de maneira geral. Em particular quanto à produção dos chamados Super-Minis computadores no país. Os jornais têm transcrito manifestações de entidades que defendem abertamente a possibilidade de formação de **joint-ventures** *de empresas estrangeiras com empresários nacionais para a produção de produtos "modernos".*

A detenção do capital majoritário em mãos de empresas nacionais não é, entretanto, garantia de domínio tecnológico. No mundo atual, o poder de decisão pertence a quem tem o domínio tecnológico. O modelo **joint-venture** *tem mostrado que, em tais associações, a tecnologia não é efetivamente transferida, sendo no máximo reimportada periodicamente. São exemplos conhecidos na indústria de telecomunicações, automobilística e naval.*

Entidades como FIESP, FIRJAN e CNI estão equivocadas quando se aliam a posições nocivas aos interesses nacionais e idênticas às manifestadas pelas multinacionais do setor; em particular pela IBM, através de seu presidente, em recente palestra promovida pela SUCESU-RJ.

A Secretaria Especial de Informática-SEI, órgão do Conselho de Segurança Nacional, partidos políticos, entidades de representação de profissionais e empresários nacionais do setor, como a Sociedade Brasileira de Computação—SBC e a Associação Brasileira das Indústrias de Computadores e Periféricos-ABICOMP, por outro lado, assumem publicamente posições firmes na defesa dos interesses nacionais, quanto à manutenção e ampliação da Política de Reserva de Mercado.

A APPD-RJ considera importante para o país e para os profissionais de processamento de dados o domínio da tecnologia no setor de informática, para o qual é fundamental a política de Reserva de Mercado.

Acreditamos que esta tecnologia é do interesse de todos os cidadãos brasileiros e que a política do setor deve ser controlada pela sociedade como um todo e garantida através de legislação aprovada pelo Congresso Nacional.

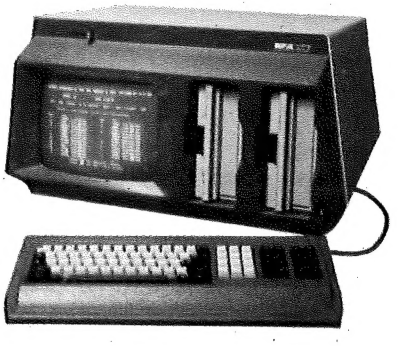
COMPUTADOR COBRA AGILIZA SERVIÇOS GRÁFICOS

Até mesmo pequenas gráficas já podem se beneficiar da moderna tecnologia para aumentar a eficiência dos serviços de fotocomposição.

O microcomputador COBRA 305 pode ser ligado às fotocompositoras existentes no mercado e, utilizando os recursos de seu Sistema Processamento da Palavra — SPP, ser usado para a geração de textos a serem fotocompostos.

O processo consiste, basicamente, na preparação dos textos no COBRA 305 através do SPP. Os textos são digitados, corrigidos e arquivados em discos flexíveis, permanecendo disponíveis para quaisquer novas alterações.

Os textos, em sua forma final, são transmitidos para a fotocompositora, que os recebe já formatados e justificados, em imagem de impressão, agilizando todo o processo da gráfica.



Esta solução tem a vantagem adicional de permitir o uso do computador para aplicações administrativas, durante o tempo em que não estiver sendo utilizado para fotocomposição.

Analisar os problemas e benefícios que a informática pode proporcionar à população, divulgar as possibilidades de sua aplicação, permitir ao leigo maior contato com os computadores e, a partir daí, o posicionamento do homem frente à tecnologia são os principais objetivos, visados pela INFORMÁTICA 83 – XVI Congresso Nacional e III Feira Internacional de Informática, que serão realizados de 17 a 23 de outubro próximos, no Parque Anhembi de São Paulo. Para o presidente do certame, Salvador Perrotti, "o encontro deste ano permitirá o contato dos usuários futuros com o computador, dentro do próprio tema geral, INFORMÁTICA A SERVIÇO DA SOCIEDADE: PRESENTE E FUTURO".

Para Salvador Perrotti, "a escolha do tema representa, também, uma abertura para a análise da situação atual, vendo-se aplicações específicas da informática dentro de campos importantes como a medicina, a educação, a engenharia e o direito. Queremos mostrar o computador a todos os públicos como uma ferramenta de todas as atividades profissionais. A ocasião será própria, ainda, para se debater as conseqüências da disseminação da informática no Brasil e a comparação de seu estágio de evolução com o de outras nações. Informática 83 terá também, segundo Salvador Perrotti, uma série de sessões plenárias, com a presença de cientistas e autoridades, nacionais e internacionais, para discutirem temas como política de informática, informática e a sociedade, informática e o emprego, informática na América Latina e o satélite brasileiro e a informática.

PARTICIPAÇÃO ESTUDANTIL

Os organizadores da Informática 83 manifestaram-se também preocupados em ampliar a programação para participação estudantil que, anteriormente, atingia apenas as áreas relacionadas ao setor. Agora, eles se voltarão para universitários dos diversos campos de atuação, alunos do segundo grau e até crianças do curso primário. Um exemplo dessa participação será o circo que será montado no Anhembi para possibilitar um divertido contato da criança com a máquina.

A programação do congresso terá também cursos e atividades específicas para educadores e pedagogos, com o objetivo de estimular o debate sobre condições e conseqüências da aplicação da informática no ensino. As atividades destinadas aos vários setores profissionais começarão com a seleção de trabalhos técnicos enviados até 31 de maio à secretaria geral.

FEIRA

A III Feira Internacional de Informática, cuja organização está a cargo da ABICOMP – Associação Nacional da Indústria de Computadores e Periféricos – pretende apresentar produtos da indústria nacional e do exterior, desde micros até computadores de grande porte, suprimentos e software. Até agora, 10 dos 22 mil m^2 da área total do Palácio das Exposições do Parque Anhembi já estão vendidos, com um total de 120 expositores. No mesmo local, será reservada uma área para a mostra de universidades e centros de pesquisa, com a exposição de produtos desenvolvidos por estudiosos e cientistas brasileiros.

Para complementar a programação dos certames e fixar sua posição dentro da Comunidade uma série de promoções será organizada, como um curso para jornalistas, destinado a esclarecer os profissionais sobre o assunto, e instalação de quiosques em áreas de grande densidade popular, onde robôs serão preparados para testes, jogos e brincadeiras. Será, também, criado um prêmio Imprensa. Informática 83 tem sua secretaria geral à Avenida Paulista 1159, 14º andar, conjuntos 1404/1405, telefone 288-9452.

SETOR DE INFORMÁTICA GANHA NOVA EMPRESA

Foi eleita em 6/06/83 a primeira diretoria da recém-criada CETUS INFORMÁTICA S. A., que vai produzir "redes locais" baseadas em projeto desenvolvido por técnicos brasileiros. Trata-se de um sistema que permite a interligação de vários equipamentos – computadores de todos os tipos, discos, impressoras, terminais etc – com a possibilidade de comunicação simultânea e compartilhamento de recursos.

A nova empresa tem um capital social de 30 milhões de cruzeiros e sua diretoria está assim constituída: Gilberto Job, presidente; José Grimberg, diretor superintendente e Jones Gracia, diretor técnico. A CETUS produzirá seus equipamentos no parque industrial de sua coligada COENCISA INDÚSTRIAS DE COMUNICAÇÕES S.A., em Brasília.

BAGÉ NA ERA DA INFORMÁTICA

A ADVANCING – PRODUTOS E SERVIÇOS EM INFORMÁTICA LTDA está com nova filial na fronteira, colocando a sua experiência e eficiência na área de Informática ao alcance de toda a população de Bagé e arredores.

Estamos ampliando os horizontes de uma empresa já conhecida e credibilitada em todo o país, por ser voltada exclusivamente à era de Informática.

Em Bagé, estarão à venda e em exposição os mesmos equipamentos vendidos na Matriz (Porto Alegre), como por exemplo: SCHUMEC, CP200, TK85 e outros.

Enfim, uma filial com todo o respaldo de sua Matriz.

O endereço da ADVANCING BAGÉ é Rua Professor Otávio Hipólito, 1740 – Telefone 2-2433.

consultoria

Trata-se de uma coluna mensal destinada a formar um elo entre o usuário e o fabricante de equipamentos abrindo-se ao leitor para recebimento de cartas que contenham perguntas sobre funcionamento, operação, preços, modificações e falhas em microcomputadores e periféricos de fabricação nacional ou importados.

Se você tem dúvidas sobre algum assunto ligado à informática, escreva para:

CONSULTORIA — REVISTA "INTERFACE" — ESTRADA DO TINDIBA, 2380 — JACAREPAGUÁ — RJ — CEP 22700.

P Como eu poderia saber quantos *K* tem um computador, ou melhor, se um computador tem 16 Kbytes de memória como posso saber se ele tem realmente 16 Kbytes? Há alguma instrução que demonstre sua capacidade de memória?

R Para você saber quantos *K* de memória tem o seu computador, por exemplo, escreva o seguinte no vídeo: ***PRINT MEM.*** Aperte a tecla ***RETURN.*** Você está pedindo ao computador para imprimir no vídeo a quantidade de memória disponível para programação. O computador responderá, em caso de 16K: 15570.

Se o seu computador for um TK82-C, TK85, NZ8000, CP200 ou TIMEX SINCLAIR 1000 apresentamos o seguinte programinha que imprimirá no vídeo os *K* de memória:

```
10 POKE 18000,33
20 POKE 18001,11
30 POKE 18002,0
40 POKE 18003,57
50 POKE 18004,68
60 POKE 18005,77
70 POKE 18006,201
80 PRINT "Os K de sua memória são:"
90 PRINT (USR (18000) — 16373)/1024; "K"
```

Em último caso, consulte o manual do fabricante do seu computador.

P Quando listo determinados programas no meu DGT-100 as variáveis A$ aparecem como nulas, mas quando rodo o programa aparecem na tela várias instruções. Como isso é feito?

R Normalmente, coloca-se ***FC*** em hexadecimal ou 252 em decimal na linha que não deve ser listada. Quando o computador encontra ***FC*** não lista nada além de ***FC.*** Aguarde um artigo nos próximos números que explicará com mais detalhes como deve-se proceder nesta situação.

P O que são programas ***compiladores*** e ***montadores*** e quais as suas diferenças?

R Os programas montadores e compiladores pertencem à família de programas que traduzem os comandos do programador para uma linguagem, código de máquina, que o processador possa interpretar.

As diferenças genéricas entre um ***montador*** e um ***compilador*** são relevantes. O ***montador*** é geralmente composto apenas para um determinado microprocessador ou uma linha de microprocessadores. Ele converte uma instrução simbólica do ***programa-fonte*** (programa em forma quase-inglês) numa do ***programa-objeto*** (linguagem de máquina). O objetivo básico de um montador é desenvolver meios de escrever programas em código simbólico que facilitem ao programador entendê-los e reconhecê-los. Por exemplo, é mais fácil escrever e lembrar de uma instrução como VAILOJA, onde VAI representa uma operação e LOJA um endereço de memória, do que 10111010011. Estes algarismos confundem e, para a maioria das pessoas, dificultam a lembrança dos códigos exigidos. Por que não deixar que a máquina faça a tradução? É exatamente o que faz um ***montador.***

O ***compilador*** possui um campo de ação mais amplo que o do ***montador.*** Enquanto o ***montador*** gera os comandos em código de máquina, um por um, o ***compilador*** pode gerar vários comandos de máquina a partir de um único comando simbólico de entrada. Isto significa que o programador pode gerar uma seqüência de programas de máquina relativamente longa, usando apenas poucos comandos simbólicos.

Os ***compiladores*** são escritos visando mais as categorias de aplicação do que um determinado tipo de máquina. Por exemplo, emprega-se o compilador ***FORTRAN*** em grande número de máquinas de diversos fabricantes. Uma outra vantagem dos compiladores é que se alguém aprender numa determinada máquina, ao usar outra sentirá apenas pequena diferença.

Os ***compiladores*** ocupam maior espaço na memória e levam mais tempo para processar do que os ***montadores,*** que são mais eficientes para as máquinas.

P Estou desenvolvendo uma interface entre lógica CMOS e lógica TTL; os ***drivers*** que estou usando

são os F4049 e F4050, que apresentam os seguintes problemas:

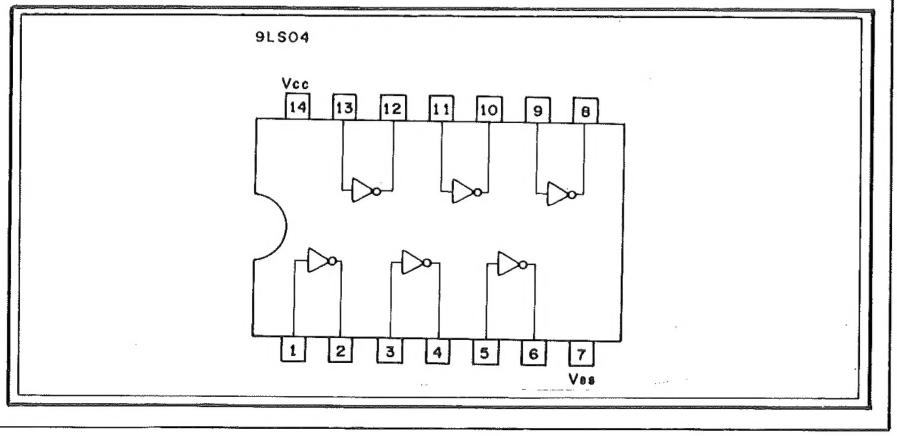
- delay de 45ns (VDD = 5V), quando preciso de 10ns máximo;
- carga máxima (load) 2 TTL standard, quando preciso de até 5;
- não permitem tensão na entrada na ordem de 10 volts.

Gostaria de saber se existe em disponibilidade no mercado nacional um chip que atenda às características do meu projeto.

R Sim. No mercado nacional, principalmente no Rio e S. Paulo, temos disponível um chip especialmente projetado para interfacear lógica CMOS com lógica TTL. Trata-se do *9LS04-Hex Inverter*, que apresenta as seguintes características técnicas:

- delay típico = 5ns;
- carga TTL (load) = 5;
- Vcc = 5 volts;
- tensão de entrada máxima = 11 volts.

O diagrama dos pinos é apresentado a seguir: ■

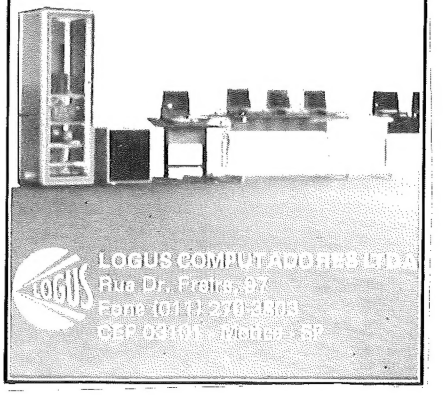


LOGUS III

A TECNOLOGIA BRASILEIRA

- CPU com 64K a 16 Megabytes de RAM
- 8 canais de comunicações
- Até 32 discos fixos ou removíveis
- Até 15 programas ativos em multi-programação
- Sistema operacional baseado em mapeamento (páginas de 4 Kbytes cada) sem "wait state"

Kapram



LOGUS COMPUTADORES LTDA
Rua Dr. Freire, 97
Fone: (011) 270-2588
CEP 03101 - Mooca - SP

ESCOLAS INTERNACIONAIS
DESDE 1891
CURSOS DE QUALIFICAÇÃO PROFISSIONAL

NOSSOS CURSOS SÃO CONTROLADOS PELO NATIONAL HOME STUDY COUNCIL •
(•) Entidade norte-americana para controle do ensino por correspondência.

provep 21583

ELETRÔNICA, RÁDIO e TV

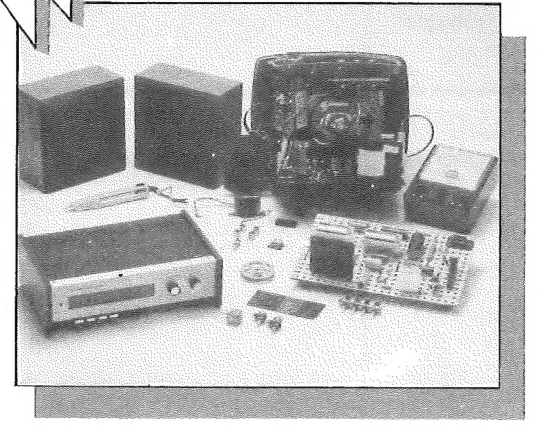
Grátis! EQUIPAMENTOS

Cursos rápidos, fáceis, eminentemente práticos, preparados pelos mais conceituados engenheiros de indústrias internacionais de grande porte.

MILHARES DE ESPECIALISTAS EM ELETRÔNICA BEM SUCEDIDOS

P/B e a cores! AM, FM, Estéreo

A teoria é acompanhada de 6 kits completos, para desenvolver a parte prática:
kit 1 – Conjunto básico de eletrônica
kit 2 – Jogo completo de ferramentas
kit 3 – Multímetro de mesa, de categoria profissional
kit 4 – Sintonizador AM/FM, Estéreo, transistorizado, de 4 faixas
kit 5 – Gerador de sinais de Rádio Freqüência (RF)
kit 6 – Receptor de televisão.



PEÇA NOSSOS CATÁLOGOS GRÁTIS
EI - Escolas Internacionais
Caixa Postal 6997
CEP 01.051 - São Paulo - SP

O curso que lhe interessa precisa de uma boa garantia!
As ESCOLAS INTERNACIONAIS, pioneiras em cursos por correspondência em todo o mundo desde 1891, investem permanentemente em novos métodos e técnicas, mantendo cursos 100% atualizados e vinculados ao desenvolvimento da ciência e da tecnologia modernas. Por isso garantem a formação de profissionais competentes e altamente remunerados.

Não espere o amanhã!
Venha beneficiar-se já destas e outras vantagens exclusivas que estão à sua disposição. Junte-se aos milhares de técnicos bem sucedidos que estudaram nas ESCOLAS INTERNACIONAIS.
Adquira a confiança e a certeza de um futuro promissor.

ENVIE CUPOM OU CARTA, HOJE MESMO!
E receba, grátis, o livreto
Como Triunfar na Vida

ESCOLAS INTERNACIONAIS
Caixa Postal 6997
CEP 01.051 - São Paulo - SP

IN-8

Enviem-me, grátis e sem compromisso, o magnífico catálogo completo e ilustrado do curso abaixo, com o livreto **Como Triunfar na Vida.**

Eletrônica

Nome. . . . . .
Rua. . . . . nº. . . . . .
CEP. . . . . Cidade. . . . . Estado. . . . . .

EI ESCOLAS INTERNACIONAIS
Sede: Av. dos Autonomistas, 791 - Salas 2 e 3
V. Yara - CEP 06.000 - Osasco - SP.

hard house

TÉCNICAS DE TROUBLESHOOTING EM CIRCUITOS DIGITAIS

por Cesar da Costa

2º PARTE

A fim de uma pesquisa de defeito (troubleshooting) eficiente é importante conhecer os tipos de falhas encontradas nos circuitos digitais. Elas podem ser classificadas dentro de duas classes principais: aquelas causadas por uma falha interna ao chip e aquelas causadas por uma falha no circuito externo ao chip.

Existem quatro tipos de falhas que podem ocorrer internamente ao chip:

- *uma ligação aberta em uma das entradas ou saídas;*
- *um curto entre uma entrada ou saída e Vcc ou Terra;*
- *um curto entre dois pinos (sendo que nenhum dos dois são Vcc ou Terra);*
- *uma falha no comando do circuito interno do chip.*

Além desses quatro tipos de falhas internas ao chip existem mais quatro falhas que podem ocorrer nos circuitos externos ao chip:

- *um curto entre um nó e Vcc ou Terra;*
- *um curto entre dois nós (sendo que nenhum dos dois são Vcc ou Terra);*
- *um caminho de sinal aberto;*
- *uma falha de um componente analógico.*

Antes de discutirmos como descobrir cada uma dessas falhas podemos descobrir o efeito de cada uma sobre a operação do circuito. A primeira falha interna ao chip mencionada foi uma conexão aberta em cada uma das entradas ou saídas. Essa falha tem um efeito diferente, dependendo se é uma conexão aberta de saída ou uma conexão aberta de entrada. No caso de uma conexão aberta de saída, figura 1, as entradas alimentadas por aquela saída estão à direita, flutuando. Nos circuitos TTL e DTL um flutuamento de entrada eleva-se para aproximadamente 1,4 a 1,5 volts e usualmente tem o mesmo efeito na operação do circuito como um estado alto. Portanto, uma conexão de saída aberta pode fazer com que todas as entradas alimentadas por aquelas saídas flutuem num "estado indefinido" visto que 1,5 volts é menor que o limiar de 2,0 volts para o "estado alto" e maior que 0,4 volts, limiar do "estado baixo".

Em circuitos lógicos TTL e DTL uma flutuação de entrada é interpretada como um estado alto. Portanto, o efeito pode ser aquele em que as entradas respondem a esse estado como se este fosse um estado alto constante.

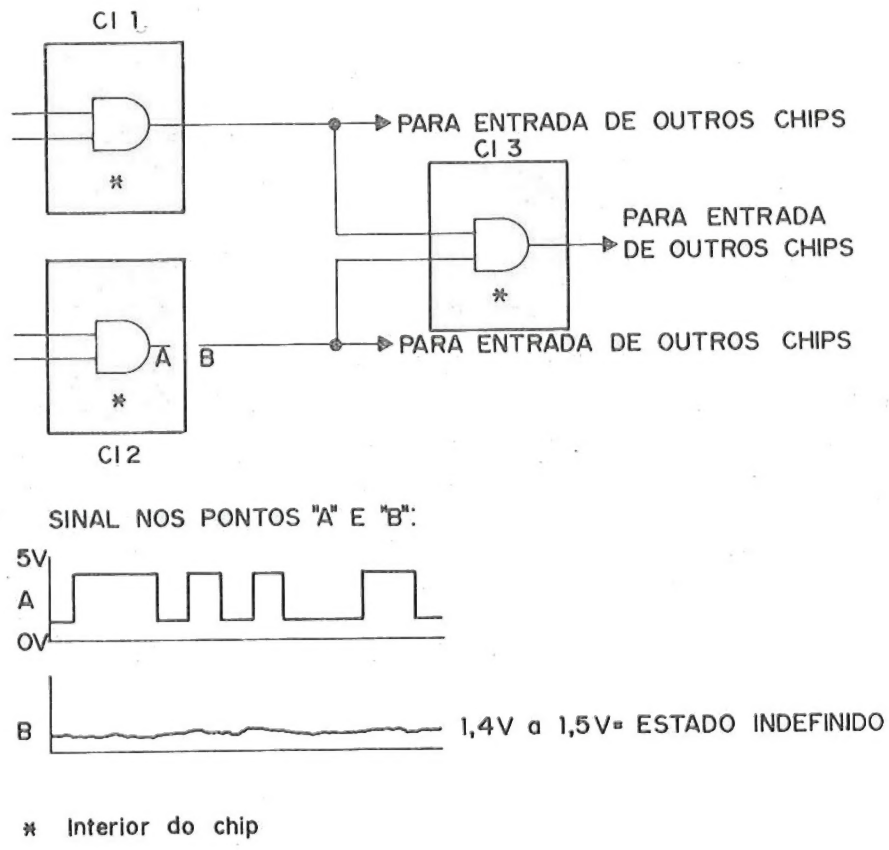


FIGURA 1 – CONEXÃO DE SAÍDA ABERTA NO INTERIOR DO CHIP

A figura 1 apresenta o resultado de uma conexão interna de saída aberta na operação do circuito. Este tipo de falha permite que todas as entradas alimentadas por aquela saída flutuem para um "estado indefinido". Esse nível é usualmente interpretado pela entrada da porta CI3 como um estado "alto". Portanto, as entradas alimentadas por uma conexão de saída aberta respondem como se um sinal constante "alto" estivesse sendo aplicado.

No caso de uma conexão de entrada aberta, figura 2, o circuito aberto bloqueia o sinal proveniente do CI2. A entrada do CI3, ponto B da figura 2, está flutuando, e o chip responde como se um estado alto constante estivesse sendo aplicado à sua entrada. É importante observar que a abertura ocorre na entrada, dentro do invólucro do chip. O sinal digital que alimenta esta entrada não será afetado pela abertura e poderá ser detectado no próprio pino do chip, ponto A da figura 2. Porém, como o sinal é bloqueado dentro do chip a saída resultante responderá como se na entrada estivesse sendo aplicado um estado alto constante.

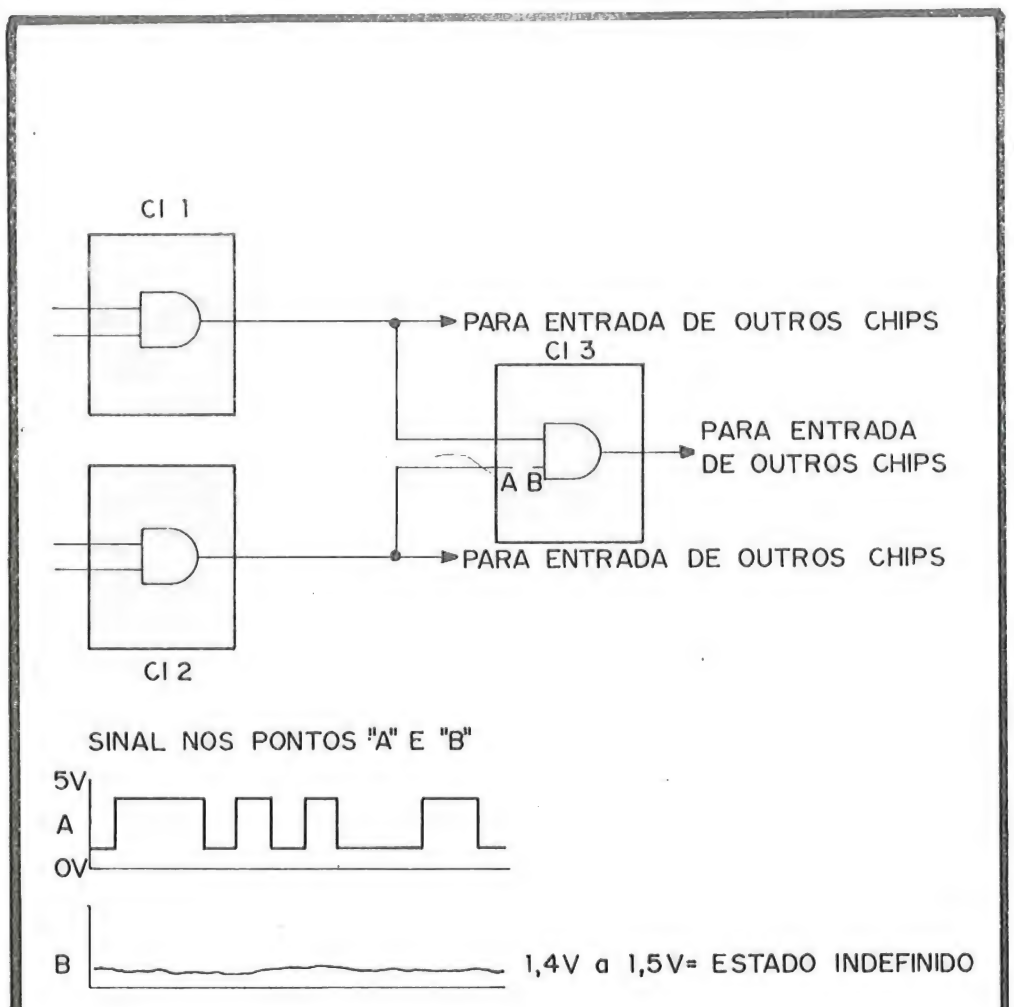


FIGURA 2 – CONEXÃO DE ENTRADA ABERTA NO INTERIOR DO CHIP

Um curto interno entre uma entrada ou saída e Vcc ou Terra tem o efeito de "amarrar" todas as linhas de sinal conectadas àquela entrada ou saída. Permanece a linha de sinal em estado alto em caso de um curto para Vcc ou em estado baixo no caso de um curto para a Terra, figura 3.

Em muitos casos isso pode causar expectativa de sinal ativo em pontos além do curto, o que é catastrófico pela duração da operação do circuito nestas condições.

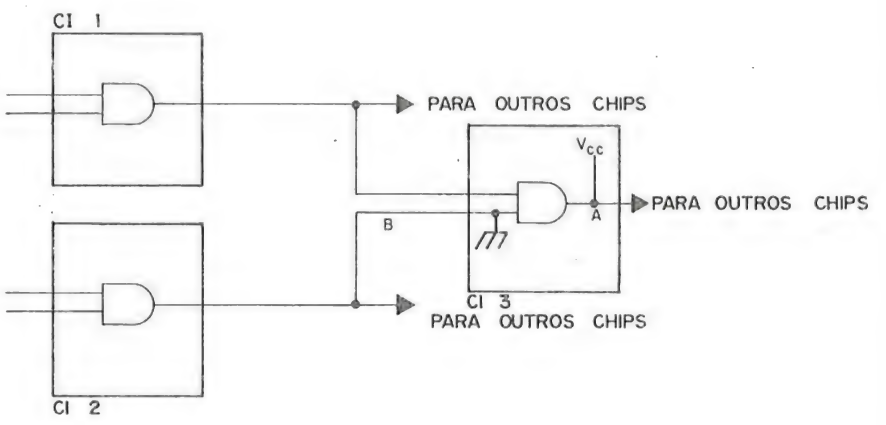


FIGURA 3 – CURTO INTERNO ENTRE UMA ENTRADA OU SAÍDA E Vcc OU TERRA

Um curto interno entre dois pinos não é para ser necessariamente analisado como um curto para Vcc ou Terra. Quando dois pinos estão em curto as saídas que os alimentam estão diretamente ligadas, ponto A e B da figura 4. Nesta situação, se as saídas A e B estão simultaneamente em estado alto ou baixo, não há nenhum problema; porém, se uma saída é alta e a outra baixa, ou vice-versa, prevalece o nível baixo nos pinos em curto, entrada do CI3.

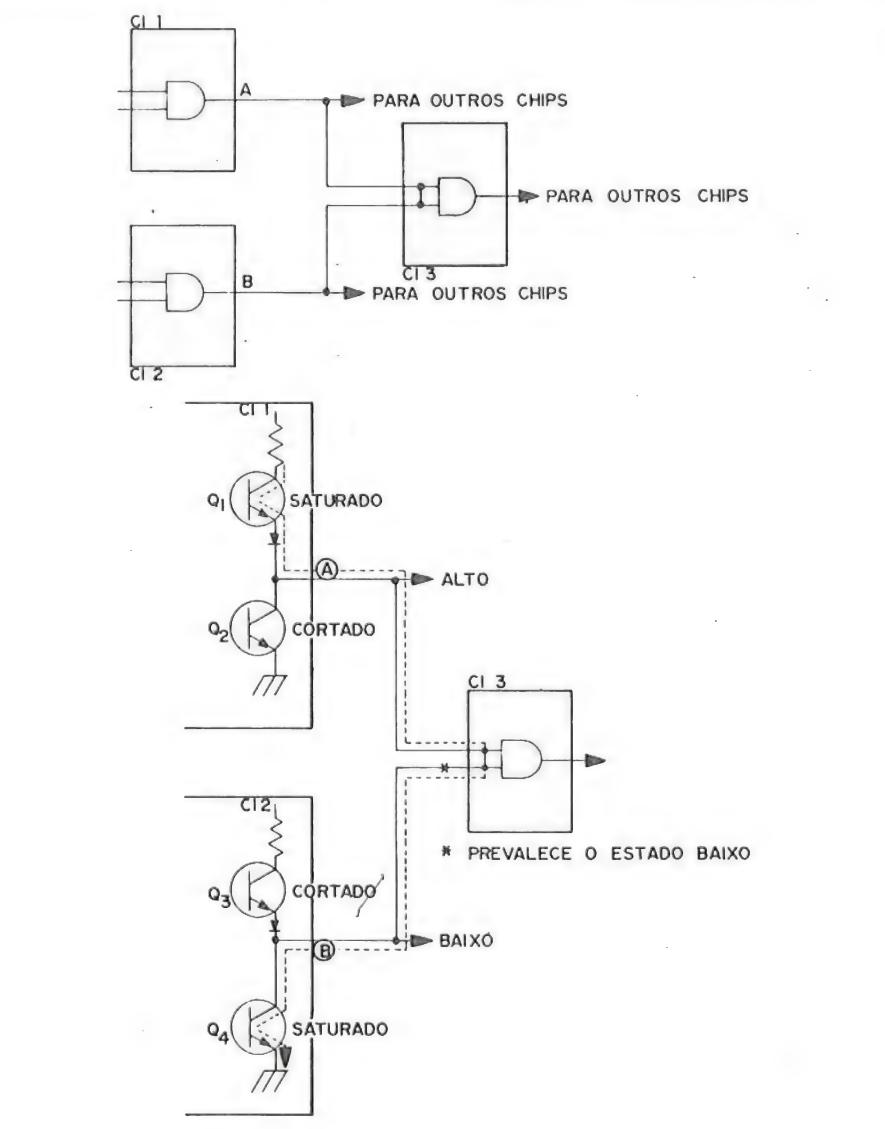


FIGURA 4 – CURTO INTERNO ENTRE DOIS PINOS

A quarta falha interna do chip é uma falha do circuito interno, transistor Q_2 da figura 5. Seu efeito é tornar permanentemente saturado o transistor Q_3 da saída **totem polem** ou o transistor Q_4 da saída **totem polem.** Portanto, essa falha bloqueia a fluência de sinais da entrada para a saída, mantendo-a num estado fixo alto ou baixo, e tem efeitos catastróficos sobre a operação do circuito.

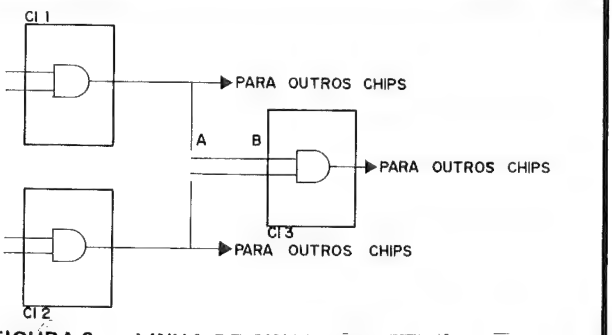


FIGURA 6 – LINHA DE SINAL ABERTA NO CIRCUITO EXTERNO AO CHIP

Um curto externo entre um nó e Vcc ou Terra é indistingüível de um curto interno ao chip. Ambos podem fazer com que as linhas conectadas ao nó sejam altas constantemente (curto a Vcc) ou sempre baixas (curto à Terra). Quando este tipo de falha é encontrado, somente uma análise física bem feita poderá revelar se a falha é externa ao chip.

Uma linha de sinal aberta no circuito tem efeito similar ao de uma ligação de saída aberta, figura 6. Todas as entradas à direita da abertura flutuam, estado indefinido, e atuam como um estado alto constante na operação do circuito. As entradas à esquerda da abertura não são afetadas e respondem corretamente à operação do circuito.

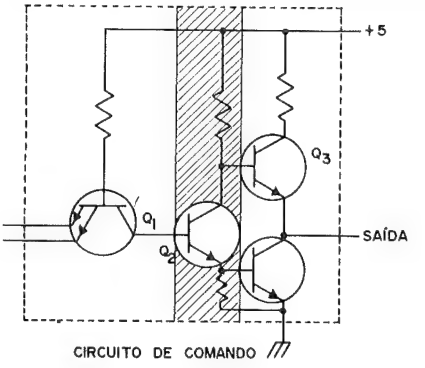


FIGURA 5 – FALHA DO CIRCUITO INTERNO DO CHIP

No próximo número veremos quais as técnicas de troubleshooting para detectar e resolver estas falhas.

micro sem programa?
soft da monk nele.

Monk, soft pronto para ser usado.

PARA D.8000, CP 500, DGT 100 e NAJA

Cadastro, Banco de Dados, Locações, Contabilidade, Contas a Pagar e Receber, Editor de Texto, Conta Bancária, Mala Direta, Visicalc, Controle de Estoque, Editor Assembler, Compiladores Basic e Cobol, jogos que ninguém é de ferro. Relação com 60 programas, todos em disponibilidade agora. Todos em português, gravados em cassette ou diskette, com manual do usuário, extremamente práticos.
Procure no seu revendedor predileto nossa relação completa de programas, ele está em condições de aconselhá-lo e dar demonstrações técnicas.
Livre-se já dos custos em ORTN's e esperas duvidosas.
Sem soft seu micro não trabalha.

monk micro informática ltda.
R. Augusta, 2690 - 2º andar - Loja 318
Tel. (011) 852-2958 - cep 01412 - SP

monk,
o software que faz você ficar feliz por ter um micro.

SOFTWARE

PARA TODO E QUALQUER TIPO DE MICROCOMPUTADORES

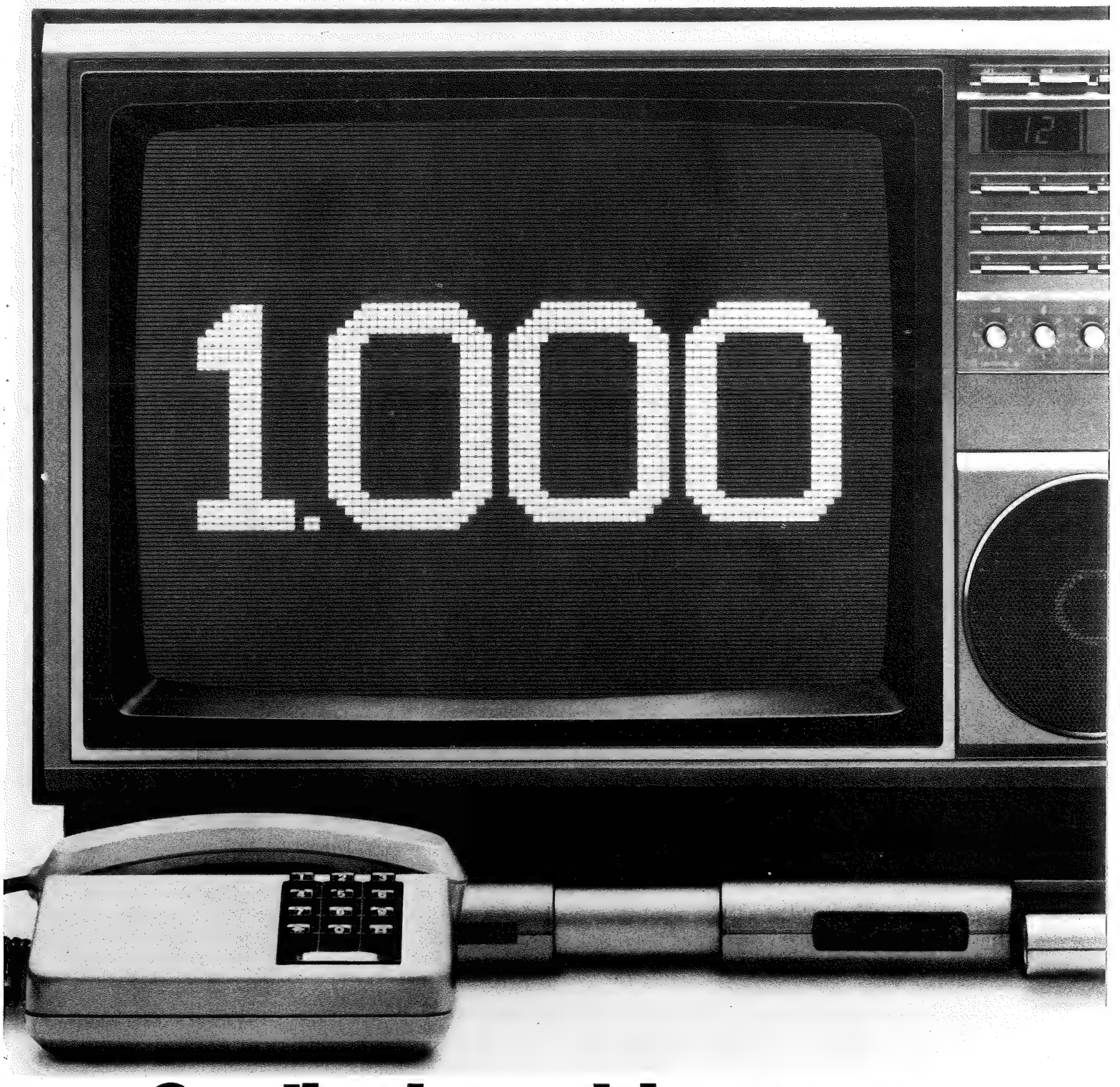
Desenvolvemos programas específicos, em fita ou diskete, para aplicações diversas.
Temos disponíveis mais de 50 programas para DGT-100, D8000, CP500, TRS80 e outros.

FINANCEIROS: Contabilidade, Controle de Estoque, Folha de Pagamento, Crediário, Faturamento, Contas a Pagar e Receber, etc.
CIENTÍFICOS: Histogramas, Gráficos, Curvas, Integral e outros.
DIVERSOS: Jogos de diversão, Video-Clubes, Mala-Direta.

☐ Descontos especiais para revendedores.
☐ Consultoria e assessoria completa na escolha do equipamento ideal e mais adequado às necessidades de sua empresa.
☐ Atendimento por reembolso postal para todo Brasil.
☐ Cursos de Basic: turmas limitadas - 10 pessoas. Duração 2 semanas. Aulas diárias (19 às 21 h.)

nasajon sistemas
Av. Rio Branco, 45 gr.1311
Tel. (021) 263-1241 -
CEP 20.090 -
Rio de Janeiro.

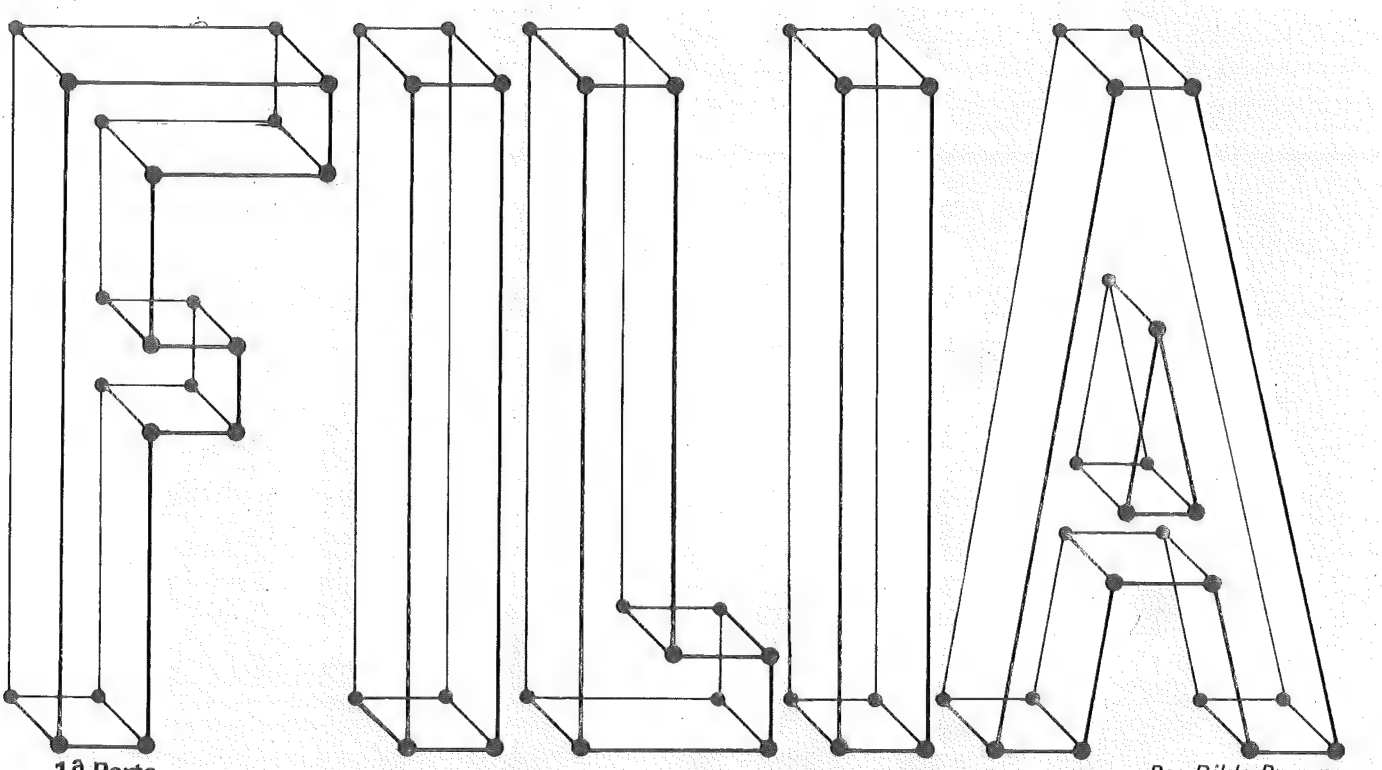
HÁ



O milésimo videotexto.

Telesp instala seu aparelho de videotexto nº 1.000. Cada terminal de videotexto colocado é um gol de nossa tecnologia no placar do progresso. A tática de juntar o telefone com a televisão deu como resultado o videotexto, filho superdotado de dois poderosos meios de comunicação. O videotexto passa pra você - sempre de primeira - as informações e serviços que levarão à meta desejada. O sistema funciona como um banco de dados, através da rede telefônica. Você escolhe o gênero de informação que deseja receber na tela do seu televisor numa operação tão simples quanto um telefonema. Videotexto não brinca em serviço. A contagem de 1.000 aparelhos prova que ele veio pra ficar. Uma vitória do pioneirismo da Telesp no jogaço do videotexto.

TELESP
TELECOMUNICAÇÕES
DE SÃO PAULO S.A.
EMPRESA DO SISTEMA TELEBRÁS



1ª Parte

Por Rildo Pragana

Nesse artigo mostraremos em várias partes como é possível escrever programas para um microcomputador sem muita capacidade de memória e com poucos periféricos. Apesar da sua simplicidade o sistema **Filia** possui um assembler residente para o 8080 ou Z-80, um editor de textos, uma linguagem estruturada de alto nível e vários utilitários que podem ser executados pelo nome digitado no teclado. Serão dadas listagens e instruções para que você mesmo consiga adaptá-lo ao seu micro. Uma versão mais sofisticada encontra-se disponível comercialmente pelo autor do referido artigo.

A parte mais ingrata do trabalho com microcomputadores está no tempo de espera para execução de programas simples, normalmente em linguagens interpretadas como o Basic. Isso sem falar no trabalho de depurar os programas e armazenar em cassete, e na impossibilidade de conjugar programas em linguagem assembly com outros em Basic. Lembre-se da última vez que você tentou entrar com programas em máquina utilizando **POKE**'s e testou-os com **USR**. Em **Filia** você poderá expandir a linguagem colocando comandos seus escritos nela própria ou em assembly (sim, existe um assembler!!!) e o sistema os tratará como se fizessem parte da linguagem original. De sobra você terá uma calculadora que lhe permitirá fazer cálculos imediatamente, bem como transformações de base numérica com a maior simplicidade.

Os números são sempre armazenados em binário, porém, comandos como HEX (hexadecimal), DEC (decimal), OCTAL ou BIN (binário) ou mesmo outros que você poderá criar farão a entrada e saída de números em qualquer base. Apesar das listagens serem dadas para o 8080/Z-80, facilmente um sistema semelhante poderá ser montado para outros processadores. Nos exemplos daremos instruções especialmente dirigidas para o TRS-80 que poderão também ser usadas no DGT-100, sem nenhuma modificação.

A linguagem **Filia** pode ser "customizada" com os comandos que você achar mais conveniente e se transformar numa linguagem aplicativa muito poderosa. Nossa linguagem é originária do **Forth**, que vem sendo usada desde o início da década de 70, inicialmente no IBM 1130. Nos Estados Unidos o **Forth** ganhou uma popularidade muito grande e muitos grupos de usuários foram criados. A **Filia** possui características semelhantes ao **Forth**, tendo sido retiradas, contudo, as maiores imperfeições do **Forth** e projetado um sistema operacional à altura da linguagem. Atualmente várias versões existem; entre elas a da Universidade Federal de Pernambuco, onde um setor desenvolve equipamentos programados em **Filia** para controle de processos. Nossa linguagem tem em comum com o **Forth** a implementação, conhecida como "código alinhavado", que você aprenderá brevemente como construir, entender e modificar. Mãos à obra.

FILIA COMPARADA ÀS LINGUAGENS TRADICIONAIS

Filia é uma linguagem parcialmente compilada, parcialmente interpretada. Falamos de linguagem compilada quando existe um programa (compilador) que traduz o texto que nós escrevemos (programa fonte) transformando-o em algo inteligível pelo computador. Normalmente são necessários muitos recursos do computador, especialmente memória, para realizar esta tarefa.

Os interpretadores, por outro lado, são mais simples e lentos, como no caso do **Basic** que provavelmente seu computador possui. Alguns compiladores traduzem para um código intermediário que não é linguagem de máquina mas que pode facilmente ser interpretado, reunindo o que há de melhor da compilação e da interpretação. Pascal, por exemplo, é uma das linguagens mais conhecidas que usa essa técnica. A linguagem intermediária no caso do Pascal é conhecida como **código-p (p-code). Filia** usa como código intermediário um grande número de ponteiros para as funções a serem interpretadas. Isso é o **código alinhavado.** O programa responsável por essa interpretação é o interpretador interno, com nome de **NEXT** (herança do **Forth**).

Em **Filia** temos muitos pequenos compiladores, que são especializados num determinado tipo de função. Assim, por exemplo, todas as constantes são criadas pelo compilador **CONST** e este somente serve a essa finalidade. **Labels** (rótulos) e números de linha de programas não existem. Estes, porém, não são necessários devido à existência de estruturas de controle. Sub-rotinas são chamadas somente pelo nome, sem um **CALL** ou **GOSUB** que existem em outras linguagens. No princípio, quem está acostumado a programar em Basic sente alguma dificuldade, mas que poderá ser rapidamente vencida. As variáveis, constantes e sub-rotinas devem ser introduzidas antes do programa principal. Se tentarmos executar ou compilar uma função não existente ainda, teremos pronta resposta do interpretador: (**nome da função**)? A função não conhecida será indicada, seguida de um ponto de interrogação.

PRIMEIROS PASSOS EM PROGRAMAÇÃO COM FILIA

Analisaremos um programa muito simples como introdução à técnica de desenvolver programas em **Filia.** Queremos um programa para acumular números numa variável chamada **TOTAL,** que deverá ser criada.

A função que acumula os dados parciais será chamada de T+, de modo que introduzindo um número seguido de T+ teremos o **TOTAL** acrescido da quantidade introduzida.

```
0 VAR  TOTAL          (criamos a variável TOTAL)
: T+   TOTAL  +=  ;   (programa principal)
```

Para usar esse programa basta escrevermos um número seguido de um espaço e do nome T+ do programa. Em seguida um **ENTER** ou **RETURN,** dependendo do microcomputador usado (a tecla de fim de introdução).

A cada **ENTER** receberemos a resposta **OK** do computador, significando que nós estamos novamente com o controle do sistema, como o **prompt** ou **ready** do **Basic.** Em **Filia** os programas começam com dois pontos seguidos do nome do programa e ponto-e-vírgula no final da definição. As palavras devem estar separadas por um espaço pelo menos. A função += , por exemplo, é uma única palavra pois não contém espaços no seu interior. Esse é um dos pontos que mais devemos ter cuidado para não ter uma interpretação diferente por parte do computador.

O programa T+ acima funciona de maneira muito simples: a função **TOTAL** é uma variável que retorna ao endereço onde esta se encontra. Em seguida += é responsável pela soma do número dado com o conteúdo original de TOTAL. É necessário que antes de T+ tenhamos colocado um número na execução, senão teremos uma mensagem de erro do sistema: **ESTOUROU A PILHA:**

A "Pilha" é uma estrutura de armazenamento que permite acesso apenas ao último valor introduzido. É como

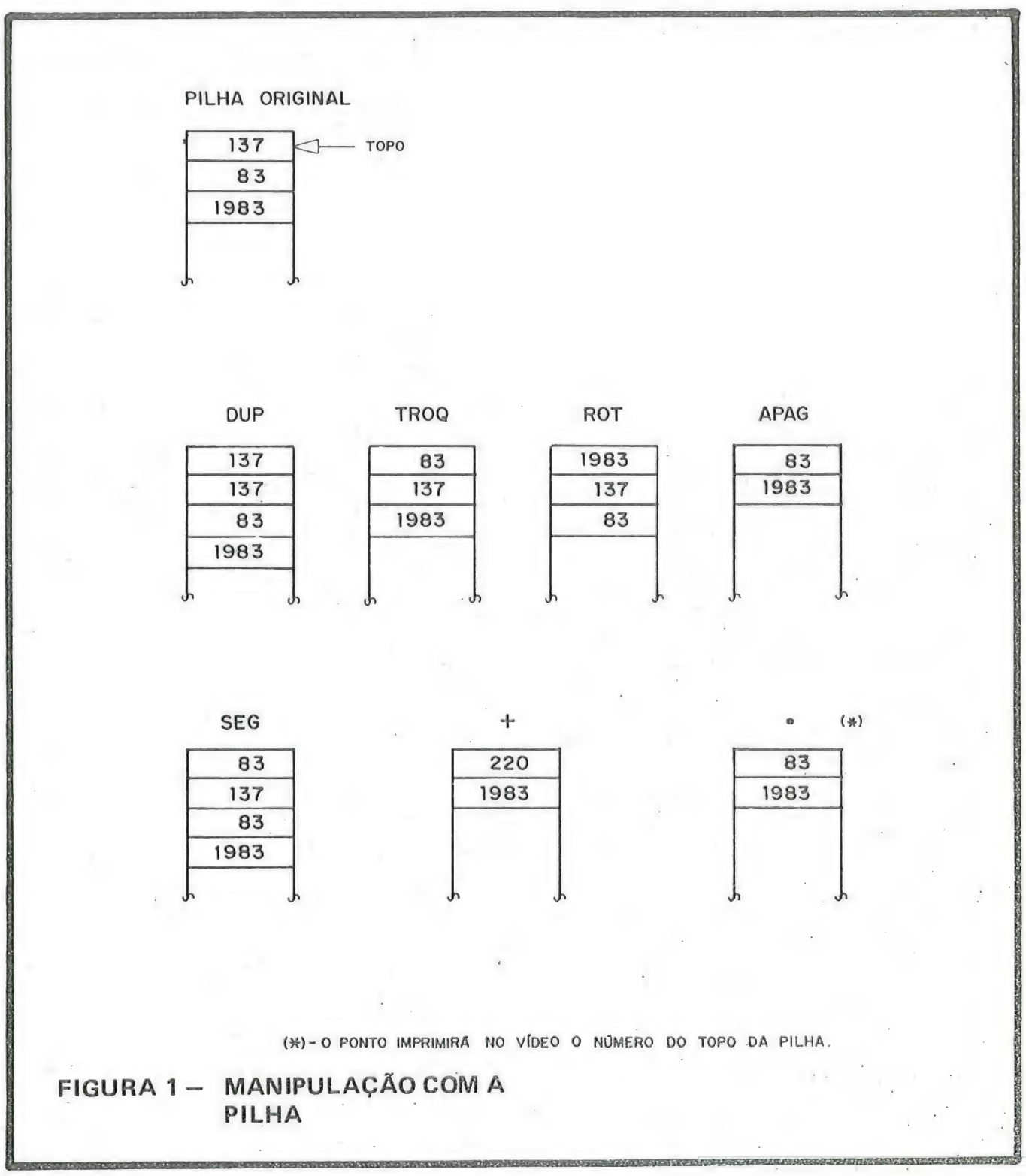


FIGURA 1 – MANIPULAÇÃO COM A PILHA

uma quantidade de automóveis que entra num beco sem saída, que só passa um carro por vez. O último a entrar será o primeiro a sair. Normalmente, "pilhas" são usadas nos computadores para guardar endereço de retorno de sub-rotinas ou salvar temporariamente registradores do processador. Em **Filia** a "pilha" normal é disponível ao usuário para os números que são introduzidos via teclado, as operações aritméticas e lógicas, bem como para passar parâmetros para as sub-rotinas (os endereços de retorno não são armazenados nessa "pilha").

Assim, no exemplo dado anteriormente, T+ espera na "pilha" um número que será acumulado em TOTAL. A execução de TOTAL aumenta um nível na "pilha" que conterá no seu topo (último item introduzido) o endereço daquela variável.

Finalmente, a execução de += retirará da "pilha" os dois itens, somando ao conteúdo de TOTAL o valor contido na "pilha" e retornando com a "pilha" vazia. A "pilha" estoura quando tentamos retirar mais itens do que ela possui, daí a mensagem de erro descrita.

Existem funções em **Filia** que são especialistas em manipular com itens da "pilha", mudando sua ordem ou criando novos itens iguais a outros já existentes. Um **DUP**, por exemplo, duplica o último item introduzido, que passa a ser dois itens idênticos na pilha. Um **SEG** funciona de modo idêntico, sendo que com o segundo item da "pilha". A função **TROQ** troca de lugar os dois últimos itens introduzidos na pilha; um **ROT** faz com que o terceiro item venha para o topo, mudando a posição dos três últimos. Um **APAG** simplesmente destrói o item do topo, diminuindo a quantidade de itens na "pilha". Um número digitado, separado por espaço de outras funções, será introduzido no topo da "pilha", efetivamente aumentando o tamanho da "pilha". Para verificarmos o conteúdo do topo da "pilha" podemos usar a função "." (ponto), que imprime o item do topo, destruindo-o. Se é interessante verificar sem perturbar a "pilha", basta fazermos **DUP** . (duplique e imprima) e o topo da "pilha" será conservado. As funções aritméticas binárias, isto é, as que usam dois operandos, têm como operandos os dois itens mais próximos ao topo, retirando-os e devolvendo o resultado da operação no seu lugar. Veja se consegue entender porque a notação polonesa é a mais adequada para a nossa linguagem.

Uma expressão escrita como (3 + 27)* 2-8 será introduzida numa expressão polonesa como 3 27 + 2 * 8 — (sem parênteses, sempre).

As variáveis, como nós já vimos, devolvem seu endereço na "pilha" e não o seu conteúdo. Para obter o valor de uma variável a função @ (pronuncie-se "valor") é usada. Essa função troca o endereço contido no topo da "pilha" pelo valor contido nesse endereço. Dessa forma **TOTAL @ .** imprimirá o valor da variável **TOTAL** quando desejarmos (não se esqueça do **ponto** se deseja ver o resultado impresso no vídeo).

Para guardarmos um certo conteúdo numa variável, ou em qualquer outro endereço do computador, usamos a função = (igual) que espera a existência na "pilha" de dois números, sendo que no topo deverá estar o endereço onde armazenar e logo em seguida o valor a ser armazenado. Isso equivale a um **assignement** no Basic, ou seja, um **LET.** Veja se consegue dizer o que restará na variável **TOTAL** após a seguinte seqüência de operações introduzidas: 3 5+ TOTAL@ + TOTAL= (responda antes de ler a próxima linha).

Os números introduzidos serão colocados na pilha e somados, com 8 como resultado. Depois o valor de TOTAL será obtido e somado com esse número; em seguida o resultado será armazenado de volta em TOTAL.

Essa mesma seqüência poderá ser substituída por 3 5 + TOTAL + =(lembra-se da função += ???).

Observando as linhas de comandos acima devemos ser capazes de entender como escrever um programa += , em função das operações com a pilha já vistas. (Tente você primeiro).

```
: +=     (definirei o programa)
  DUP    (duplique o endereço para
          não perder)
  @      (obtenha o seu conteúdo)
  ROT    (obtenha o valor a ser soma-
          do)
  +      (some-os)
  TROQ   (ponha na ordem esperada
          por = )
  =      (guarde o resultado de volta)
         (final do programa)
```

Observe a importância da manipulação com a pilha correta. Um programa deve destruir seus operandos e retornar

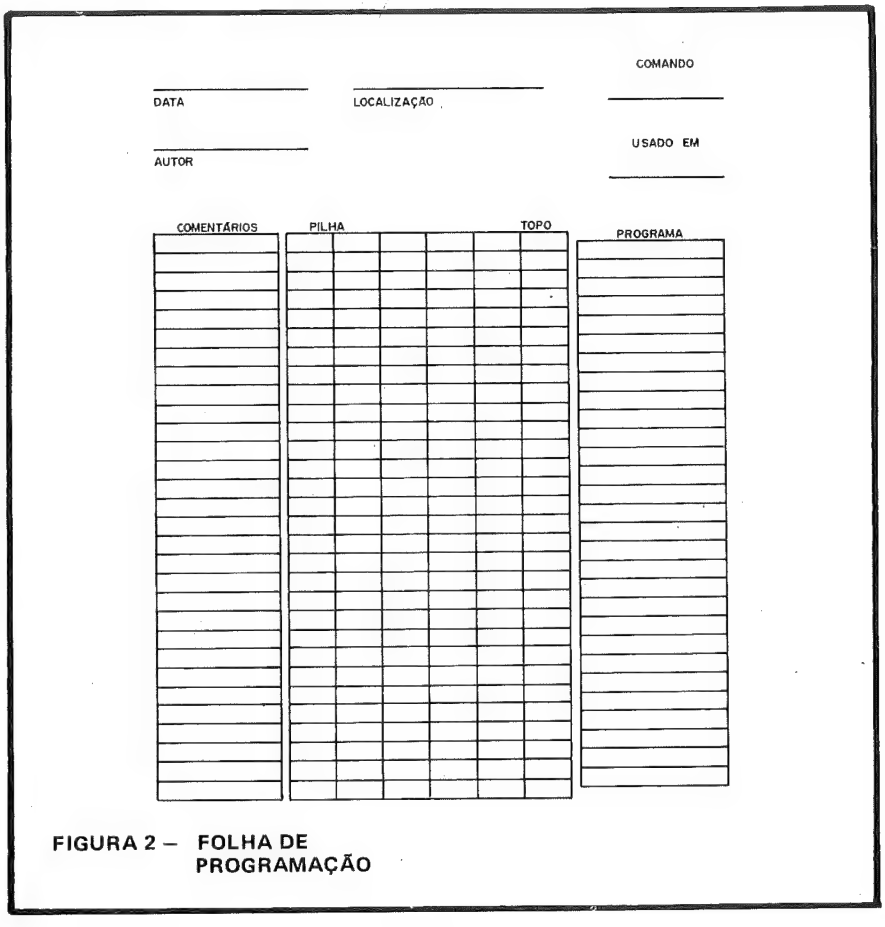


FIGURA 2 — FOLHA DE PROGRAMAÇÃO

os resultados, quando estes existirem, sem deixar "sujeiras" na pilha. Uma folha de programação como a contida na figura 2 pode ajudar muito a seguir o que acontece com a "pilha" durante a execução do programa.

SISTEMAS NUMÉRICOS

A **Filia** pode trabalhar com virtualmente qualquer tipo de sistema numérico e não apenas em decimal como a maioria das linguagens. Uma variável chamada **BASE** contém a base do sistema de numeração correntemente em uso. Evidentemente, se escrevermos BASE @ . teremos sempre como resposta um 10, que é o valor da base no sistema de numeração de mesma base: no sistema binário 10 é o número dois, em octal 10 é o número oito e assim por diante. Felizmente, existem funções que nos permitem mudar a base do sistema, o que simplifica bastante nosso trabalho.

Para fazer conversão de números entre dois sistemas de numeração basta mudarmos para o sistema desejado, introduzir o número, mudar novamente para o sistema onde queremos a conversão e executar o "ponto" para imprimir o resultado. Todos os números são armazenados na memória como binário e estas conversões para a base corrente são feitas pelo programa que introduz números e o que os imprime.

Como exemplo, veremos um programa que escreve em hexadecimal qualquer número, sem mudar a base corrente do sistema.

```
: . HEX      (ponto-hex é o nome do programa)
  BASE       (tome o valor da base corrente)
  TROQ       (troque de posição para obter o número a imprimir)
  HEX        (mude a base para hexadecimal)
  .          (imprima o número)
  BASE @     (restaure a base original do sistema)
;            (fim do programa)
```

Suponhamos que estamos com base decimal (BASE = 10) e desejamos definir os programas OCTAL, HEX e BIN. O que devemos fazer?

```
: HEX 16 BASE = ;     (programa para mudar a base para hexadecimal)
: OCTAL 8 BASE = ;    (idem, para base octal)
: BIN 2 BASE = ;      (o mesmo para binário)
```

Um fato importante é que os números introduzidos (16, 8 e 2) foram introduzidos com base decimal. Uma vez compilados esses números são constantes que não se alteram com nova mudança de base. É preciso ter muito cuidado com a base do sistema quando pretendemos compilar um programa com constantes numéricas, para termos certeza do valor do número compilado.

UM POUCO DE ENTRADA E SAÍDA

Até agora o único meio mostrado de se obter respostas do computador foi com a função "." (ponto), que imprime o topo da pilha. Algumas vezes necessitamos de mostrar um caractere ou até mesmo uma mensagem alfanumérica no vídeo, e uma linguagem prática deve possuir tais comandos. Em **Filia** podemos mostrar um caractere desde que saibamos o seu código ASCII (representação em 8 bits de um caractere). A função que realiza esta tarefa é OC (proveniente do termo **Forth: Output Character**).

Fazendo HEX 41 OC, por exemplo, teremos impresso no terminal um **A** cujo código ASCII é 41 em hexadecimal. Outra função particularmente útil é **TEC,** usada para obter o código ASCII de uma tecla pressionada. Se durante a execução de um programa está contido um **TEC** o computador ficará "parado" esperando que seja acionada alguma tecla e então retornará na pilha o seu código ASCII, continuando a execução após a ocorrência do **TEC.**

Finalmente, se quisermos que o programa imprima uma mensagen para o terminal ou o monitor de vídeo (depende do seu computador) existe a função ." (ponto-aspas) que imprime tudo que vem após sua execução (texto) até que seja encontrada outra aspas. Por exemplo, se digitarmos ." ESTA É UMA MENSAGEM" será impressa a frase ESTA E UMA MENSAGEM seguida do **prompt** OK. Como toda palavra em **Filia,** o ponto-aspas deve ser seguido de pelo menos um espaço, e o primeiro destes fará parte do nome ." e não será mostrado juntamente com a mensagem.

O programa dado a seguir pede que você digite uma tecla e então mostra qual a tecla digitada, seguido de seu código ASCII.

```
: CONVERTE ASCII   (nome do programa)
  ." BATA ALGUMA TECLA (mostra essa mensagem)
  TEC              (leia uma tecla)
  DUP              (duplique seu código para não perder)
  OC               (mostre qual a tecla digitada)
                   (e em seguida o seu código ASCII
                   (fim do programa)
```

Esse programa, quando executado, tem a desvantagem que a mensagem ficará colada ao eco da tecla e ao seu código numérico (ASCII).

Algumas funções úteis são as que permitem passar para a próxima linha, retornar para o início da mesma linha, limpar a tela, ou mesmo posicionar diretamente o cursor do vídeo para que possamos escrever em qualquer lugar da tela. Para retornar o carro para o início da próxima linha como uma máquina de escrever o faz, temos a função CR (**carriage return**), que poderíamos ter colocado várias vezes no programa anterior para separar os resultados e mensagem impressos. Para posicionamento direto do cursor podemos usar **POSCUR,** que espera na "pilha" dois números correspondentes às coordenadas X e Y respectivamente, aonde deverá ficar o cursor. **0 0 POSCUR** posiciona o cursor no canto superior esquerdo do vídeo, enquanto que **0 1 POSCUR** põe o cursor na segunda coluna da primeira linha da tela.

A seqüência 0 0 POSCUR ." PRIMEIRA LINHA" CR . "SEGUNDA LINHA" CR faz com que apareça na parte superior da tela o seguinte:

```
PRIMEIRA LINHA
SEGUNDA LINHA
OK
```

(o **OK** é o retorno do computador ao seu controle)

Rildo Pragana
SETOR DE SISTEMAS DIGITAIS — DEPTº FÍSICA UFPE — Recife — PE
Endereço para correspondência: Rua Viscondessa do Livramento, 54-A — Derby 50000 — Recife — PE

Espero que esta primeira parte tenha agradado aos leitores. No próximo número veremos como podemos controlar a execução de um programa, mesmo sem a existência de **labels**, com condicionais como SE . . . SENÃO . . . ENTÃO, INÍCIO . . . ATÉ e outras estruturas. Bem-vindos ao mundo maravilhoso de **Filia.**

BIT SLICED

Uma opção para 32 bits

4° PARTE

IMPLEMENTAÇÃO DE SEQÜENCIADORES COM FPLA (FIELD-PROGRAMMABLE LOGIC ARRAYS).

Os circuitos apresentados nas partes anteriores deste artigo são particularmente simples, porém, para situações mais práticas eles podem tornar a microinstrução bem extensa. Uma maneira de contornar essa desvantagem é alimentar os **N bits** do campo de próximo endereço da microinstrução com uma **PROM** ou **FPLA,** contendo **M bits** ($M > N$), que é aplicada ao contador de endereço de microprograma para carregamento paralelo. Isto corresponde a dois níveis de implementação de memória de microprograma: um **microstore** e um **nanostore.** Contudo, essa técnica diminui a velocidade de processamento, visto que esta implementação requer um ciclo de acesso extra e torna a microprogramação ainda mais difícil, visto que envolve o manejo de dois espaços de endereço.

Uma **FPLA** pode ser vista como uma implementação LSI da clássica estrutura **Soma-de-Produtos.** A figura 1 mostra um diagrama simplificado de uma FPLA enquanto a figura 2 apresenta mais detalhes da estrutura interna. O primeiro nível **AND ARRAY** (formação **AND**) executa a lógica **AND** para combinações selecionadas de sinais da entrada.

Estas portas **AND** executam o **produto lógico ou P-termos:** $P_1, P_2, \ldots, P_m$, onde "**m**" é tipicamente 48 ou 96.

O segundo nível **OR ARRAY (formação OR)** é composto de portas **OR** que executam a **soma lógica** de alguns ou todos os **P-termos** produzidos pelo primeiro nível, fornecendo uma saída final ou soma de termos: $F_1, F_2, \ldots, F_k$, onde **K** é tipicamente 8.

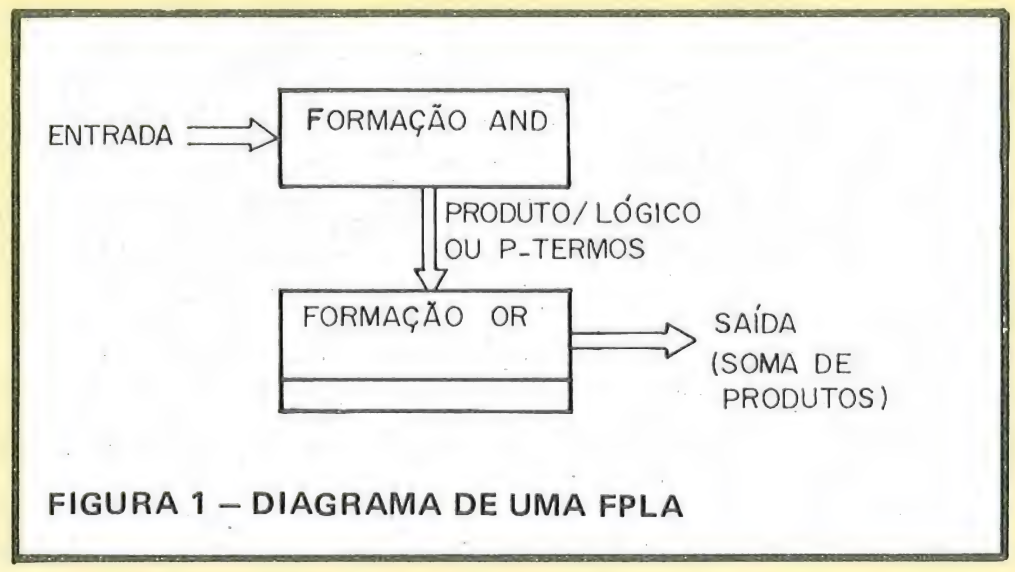


FIGURA 1 – DIAGRAMA DE UMA FPLA

Comparadas às **PROM's** (Programmable Read Only Memory) as **FPLA's** são mais eficientes porque permitem um grande número de entradas. Um chip **PROM** pode ser considerado como composto de um decodificador, que transfere os endereços de entrada para uma localização da memória, e uma parte de memória para armazenamento da informa-

ção. Enquanto a parte de memória é programável, a parte do decodificador é inalterável.

As FPLA's diferem das **PROM's** na parte relativa ao decodificador que também pode ser programável. Na implementação de circuitos combinacionais com **FPLA's** podemos ir diretamente da função lógica para a programação das formações **AND** e **OR.**

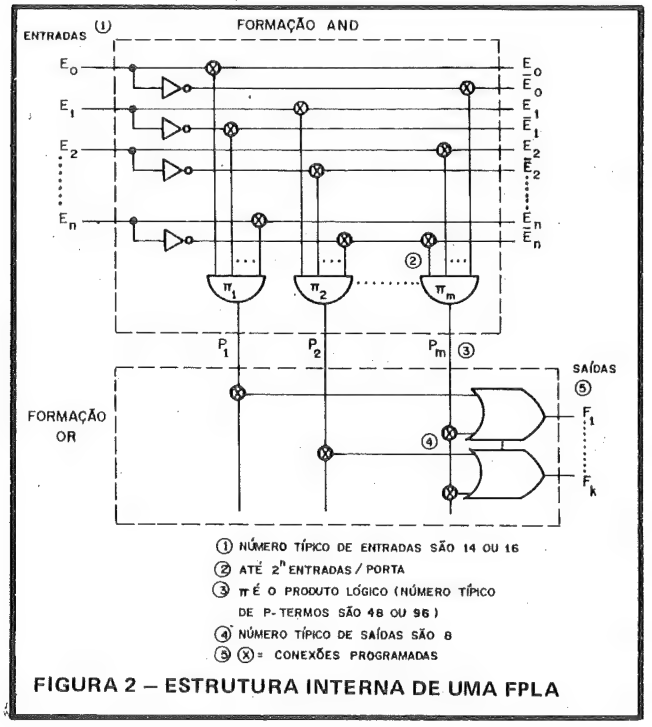


FIGURA 2 – ESTRUTURA INTERNA DE UMA FPLA

Esta vantagem torna-se mais clara nos casos em que se tem um grande número de entradas, isto é, muitas variáveis. Nesses casos, escrever uma tabela-verdade seria um trabalho árduo.

A **FPLA** é sugerida quando houver um grande número de entradas, porém, somente se um pequeno **subset** dessas combinações possíveis de entrada forem necessárias. Assim sendo, pode-se usá-la na seção do **microprocessador bit sliced** para a decodificação da macroinstrução; para decodificação do hardware e do programa de **status;** como uma memória de microprograma quando o **set** de instruções utiliza um pequeno número de palavras e longos códigos de operação; e para decodificação de microinstrução.

A figura 3 apresenta uma configuração onde as FPLA's são usadas para decodificação da macroinstrução e para decodificação do hardware e do programa de **status.**

Substituir a memória do microprograma da seção de controle por uma ou mais **FPLA's** nada mais é do que implementar uma memória somente de leitura de índice de endereço, onde os desvios condicionais são mais "vindo de" do que "indo para".

SEQÜENCIADORES COMERCIAIS DE MICROPROGRAMA LSI

Atualmente, os seqüenciadores comerciais de microprograma **LSI** possuem todas as características mencionadas anteriormente e mais algumas. Eles permitem um maior número de sinais de controle e sinais de bits de estado e possuem os requisitos necessários para implementação de funções de incrementação de endereço e complexo microdesvio condicional. Também possuem um contador de **loop** no próprio **chip** para **loops** de microprogramas repetidos e incluem "pilhas de ponteiros" (STACK POINTER) que garantem o retorno dos endereços das micro sub-rotinas. O diagrama em bloco de um seqüenciador lógico LSI comercial é apresentado na figura 4 onde o próximo endereço lógico é mais complexo que os mencionados anteriormente.

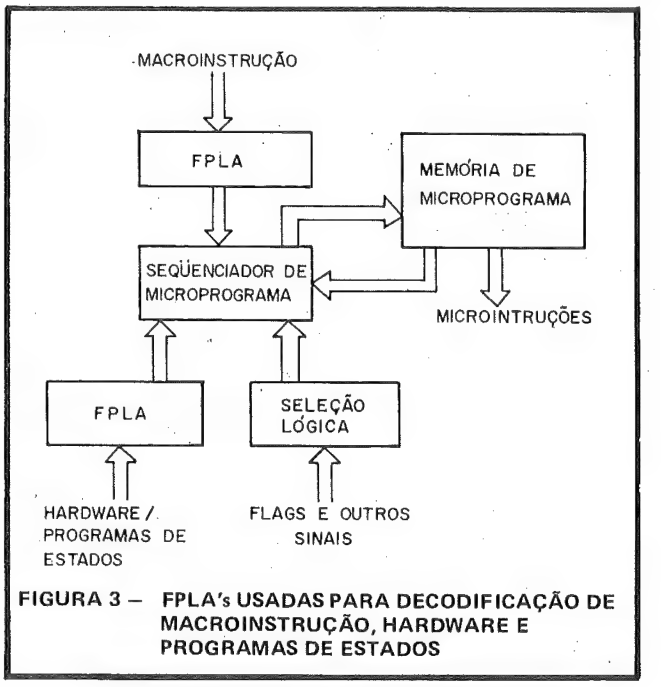


FIGURA 3 – FPLA's USADAS PARA DECODIFICAÇÃO DE MACROINSTRUÇÃO, HARDWARE E PROGRAMAS DE ESTADOS

O principal objetivo de um seqüenciador de microprograma é apresentar um endereço para a memória de microprograma, de maneira que a microinstrução possa ser buscada e executada.

Parte do próximo endereço lógico do seqüenciador determina a fonte do endereço específico a ser carregada no registro contador de endereço do microprograma. A escolha da fonte de endereço é baseada nos bits de informação do próximo endereço que o seqüenciador recebe. Geralmente esses bits podem especificar uma das seguintes informações: incremento, salto condicional da próxima instrução, desvio condicional para uma sub-rotina, **Push/Pop** na **"pilha"** e assim por diante. Muitos desses seqüenciadores LSI incluem no chip um registro acumulador, com quatro níveis de profundidade, operando na forma LIFO (Last In First Out – último a entrar, primeiro a sair). Esse acumulador é usado como armazenador temporário de endereços durante um **loop** de microprograma ou durante um desvio da micro sub-rotina, operando sob um comando tipo **Push** e **Pop.**

Os seqüenciadores de microprograma LSI, disponíveis no mercado, podem ser divididos em duas categorias: aqueles com um número fixo de posições de memória dentro de sua capacidade de endereço e aqueles que são dispositivos **bit slice** expandíveis em ligações em cascatas. Como membros da primeira categoria temos o INTEL 3001, figura 5, as MONOLITHIC MEMORIES 67110, o SIGNETICS 8 x 02, figura 7, e o FAIRCHILD 9408.

Os seqüenciadores **bit slice** são normalmente de 4 bits. Exemplos representativos são o **ADVANCED MICRO DEVICES 2909 e 2911,** figura 6 e o **TEXAS INSTRUMENTS 74LS482.** Enquanto os seqüenciadores **bit slice** são facilmente expansíveis, os seqüenciadores de endereço fixo necessitam de circuitos adicionais e técnicas de "paginação" para estender a faixa de acesso à memória.

Na categoria de endereço fixo, o INTEL 3001 e MONOLITHIC MEMORIES 67110 podem criar endereços de 9 bits e acessar diretamente 512 palavras de memória de microprograma, enquanto que o

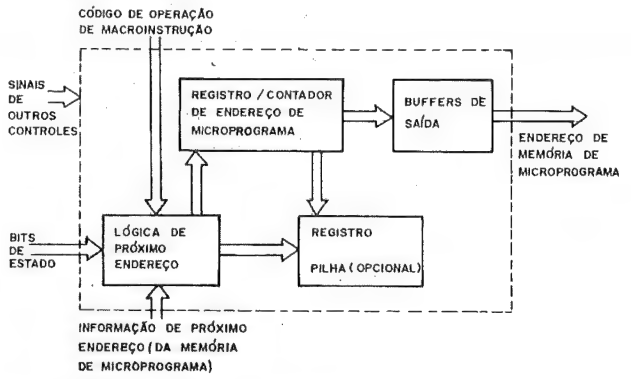


FIGURA 4 – DIAGRAMA EM BLOCO DE UM SEQÜENCIADOR LSI DE MICROPROGRAMA

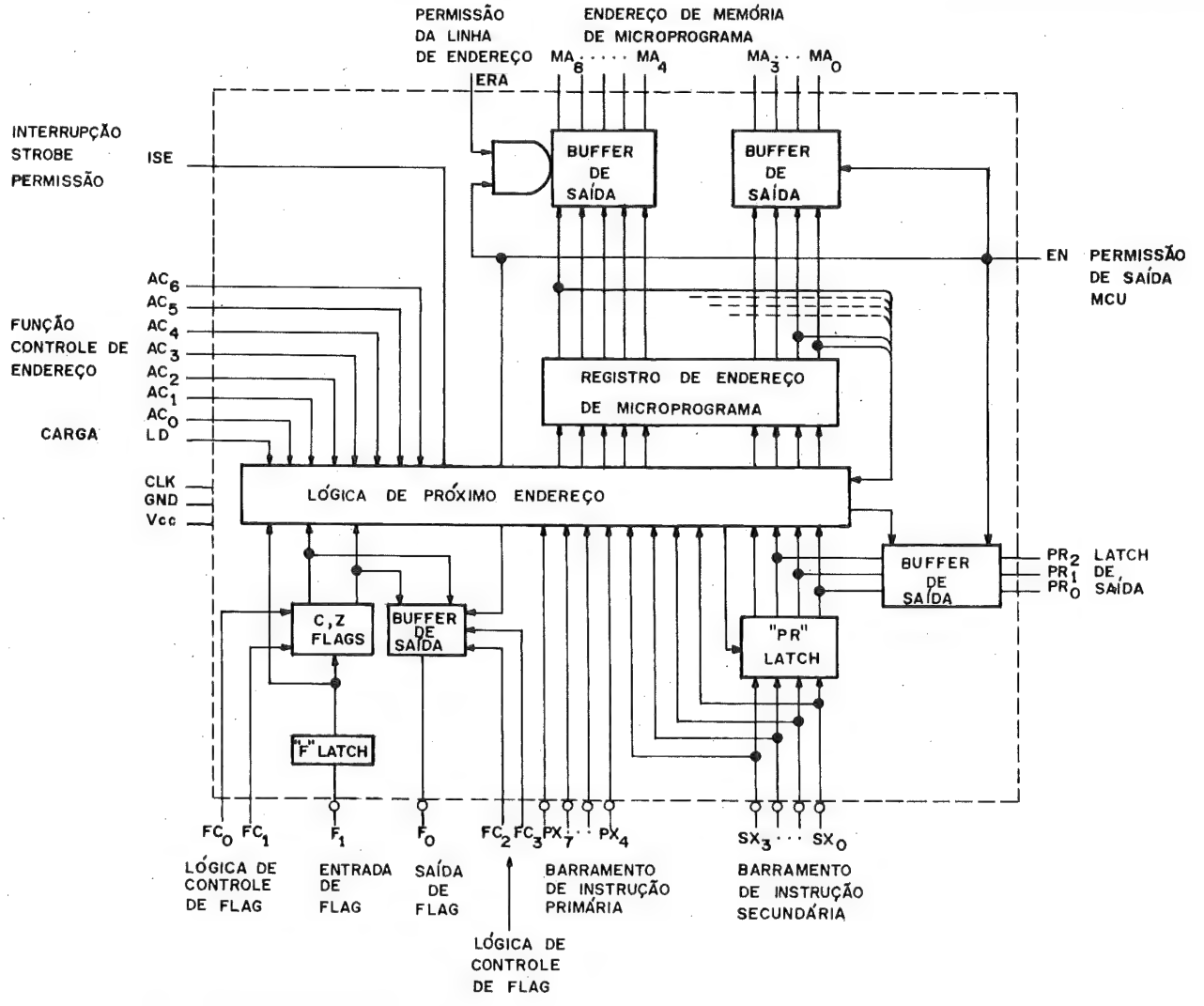


FIGURA 5 – SEQÜENCIADOR LSI INTEL 3001

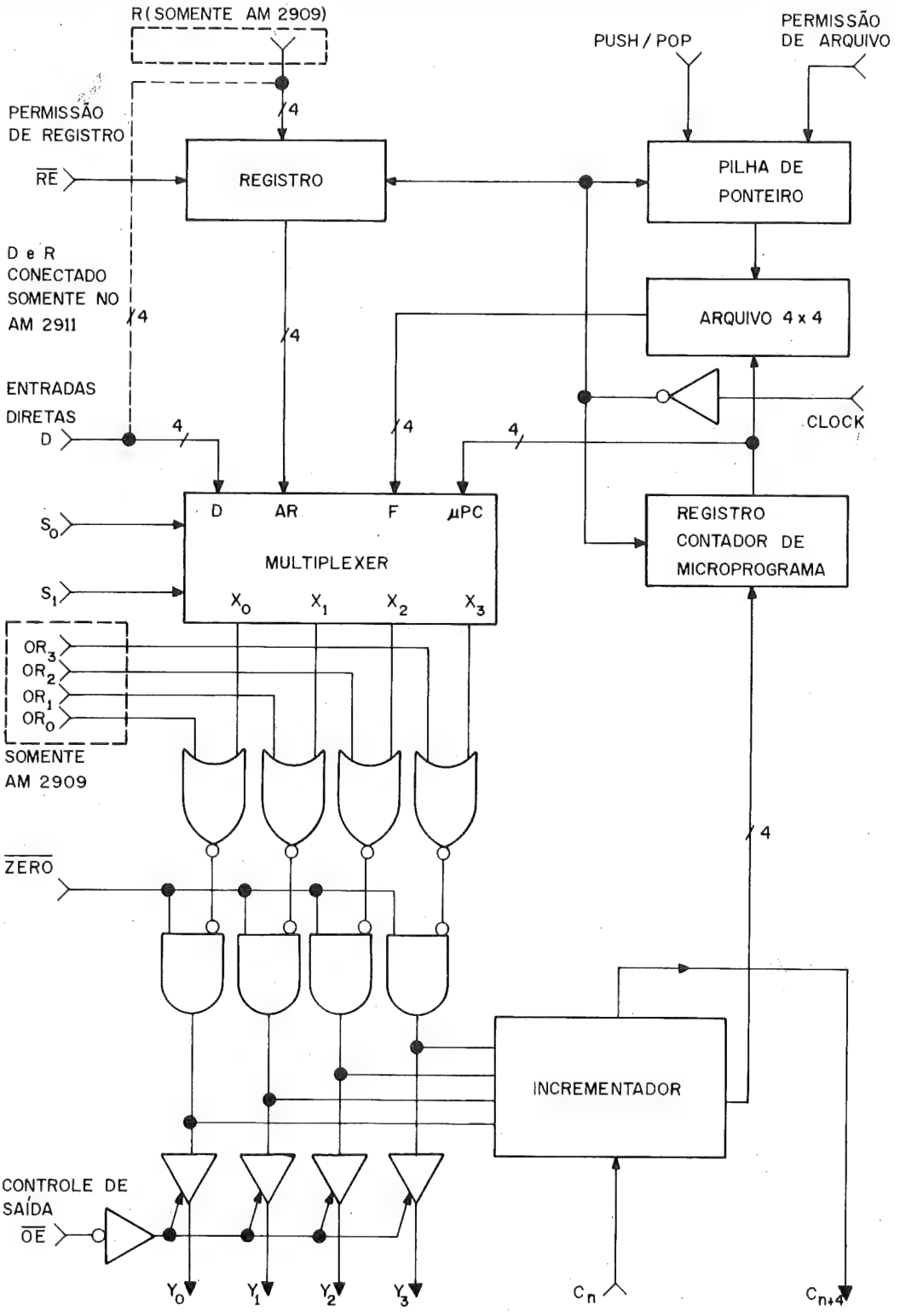


FIGURA 6 – SEQÜENCIADOR LSI AM 2909/2911

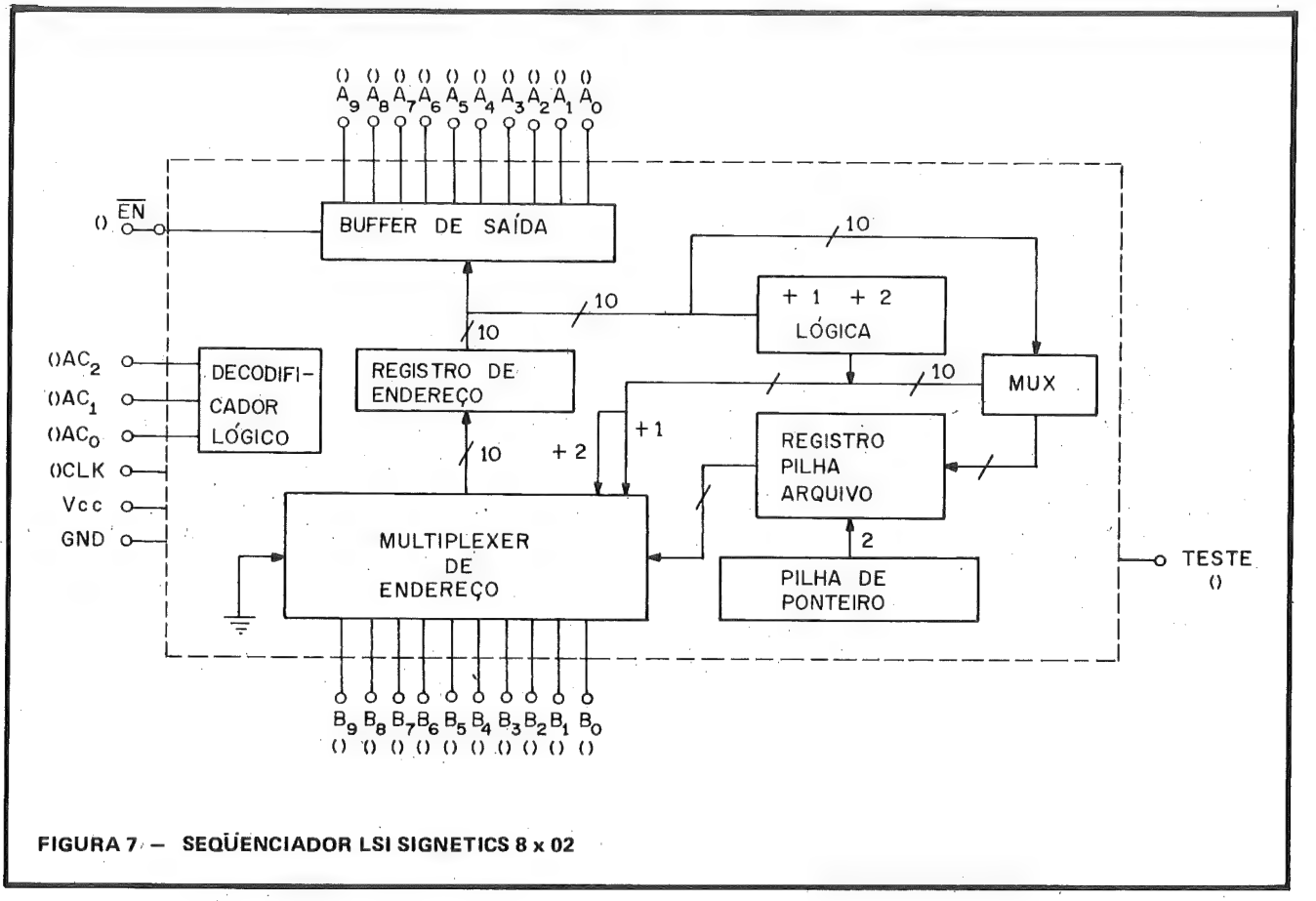


FIGURA 7 — SEQÜENCIADOR LSI SIGNETICS 8 x 02

SIGNETICS 8 x 02 e o **FAIRCHILD** 9408 podem gerar 10 bits de endereço e acessar 1024 palavras da memória de microprograma. Os seqüenciadores **bit slice** de quatro bits (2909/2911, 74S482) são ligados em cascata, o que permite, por exemplo, dois chips endereçarem 256 palavras, três chips endereçarem 4096 palavras e assim sucessivamente. Alguns seqüenciadores novos da **ADVANCED MICRO DEVICES** são o 2910, 12 bits de endereço fixo, e o 2930/2931, **bit slice** de 4 bits.

Essa facilidade de expansão torna-se importante uma vez que as memórias de microprograma e os programas de controle que elas contêm estão tornando-se cada vez maiores (vide o caso dos superminicomputadores de 32 bits).

Outra diferença entre os seqüenciadores LSI disponíveis é que vários deles (3001, 8 x 02, 67110, 9408 e 74S482) dispõem no chip de **latches** de saída de endereço, enquanto outros (2909/2911) não possuem os **latches.**

Isto permite maior flexibilidade, porém, requer ao projetista acrescentar seus próprios **latches.** Os **latches** de saída de endereço no chip permitem ao projetista utilizar uma técnica de **PIPELINING** que conduz a uma operação mais rápida, embora isto torne a microprogramação mais difícil.

Alguns dispositivos apresentam um sistema de endereçamento não-convencional. Por exemplo, o seqüenciador INTEL 3001 que trabalha com um espaço de endereço de microprograma organizado numa formação de dimensão **N.** Embora isto possa acarretar uma redução no tamanho da microinstrução e na lógica de próximo endereço, resulta em algumas perdas de capacidade de endereçamento e num aumento de esforço na microprogramação, já que o programador tem que "guardar na mente" a posição exibida no desvio de microprograma.

BIBLIOGRAFIA

- BERNARDY, A. — "Microprogrammed Multiprocessor Graphic Controller, Preprints, Sixth Annual Workshop on Microprogramming", ACM, Setembro 1973, pp. 131-133.
- BIRKNER, J. — "PLAS's: When and How to use them" — Digest of Papers, COMPCON 76 Spring, San Francisco, Califórnia, Fevereiro 24-26, 1976. pp. 160-164.
- CASAGLIA, G. F. — "Nanoprogramming VS. Microprogramming", Computer, Janeiro 1976, pp. 54-58.
- Central Processor Design Using the INTEL Series 3000 Computing Elements — Application Note AN-16, Santa Clara, Califórnia: INTEL CORPORATION.
- CHATTERGY, R. — "Microprogrammed Implementation of a Scheduler", SIGMICRO Newsletter, Vol. 7, Nº 3, Setembro 1976.
- CONN, R. B.; N. A. Alexandridis e A. Avizienis — "Design of a Fault-Tolerant, Modular Computer with Dynamic Redundancy", AFIPS conf. Proc., FJCC, 1972, pp. 1057-1067.
- CURLEY, R. — "DEVIANT: D Machine Emulation of Virtual Architecture via a Network", UCLA-ENG-7742 (ERDA: UCLA-34P214-51), Junho 1977.
- MICK, J. — "AM 2900 Bipolar Microprocessor Family", Electronics, Eighth Annual Work shop on Microprogramming, Chicago, Illinois, Setembro 1975, pp. 56-63.

POLY 201 DATA ENTRY Segurança absoluta na entrada de dados.

Fala com a linguagem Cobol, fácil de operar como um telefone.

O POLY 201 DE foi concebido para você resolver com grande facilidade e agilidade a coleta, entrada e verificação de dados. Com ele você desafoga o computador central, liberando-o da tarefa de entrada de dados, disciplinando e otimizando ainda mais sua utilização.

Características básicas do POLY 201 DE

O POLY 201 DE dispõe de linguagem POLY DEL, linguagem de alto nível que fornece ao usuário enormes facilidades de programação, além de poder-se linkeditar com programas COBOL.

A linguagem POLY DEL é composta de seis módulos básicos:

- Módulo selecionador de estado
- Módulo de formatação de telas (programação)
- Módulos de tratamento de arquivo de dados sem controle de formatos
- Módulo de execução de pesquisa, mediante chave
- Módulo de entrada de dados onde é permitida a digitação, alteração, inserção e deleção de caracteres, avançar e recuar campos, gravar, atualizar, inserir e deletar registros
- Módulo de verificação de dados.

Possui um controle total de produtividade do operador.

Com 64 kb de RAM,utiliza-se de discos flexíveis como suporte magnético, podendo-se conectar unidades de fita magnética e impressoras de várias velocidades.

Dispõe de recursos de comunicação de dados através de ligações síncronas ou assíncronas, conectando-se a diversos computadores.

Possui teclado especialmente projetado para a entrada de dados.

POLY 201 DE, uma grande conquista na entrada e processamento de dados.

ARU
ASSISTÊNCIA PRÉ-PÓS VENDA SISTEMAS E SERVIÇOS POLYMAX

A Polymax presta a sua homenagem a todos os homens que fizeram conquistas geniais, melhorando a qualidade de vida da humanidade. Como Graham Bell, o pai do telefone.

Polymax
SISTEMAS E PERIFÉRICOS S.A.
Uma grande conquista

MATRIZ: PORTO ALEGRE (RS) - FONE: 42-7833.
FILIAIS: ABC (SP) - FONE: 454-4922 - **BELO HORIZONTE** (MG) - **BRASILIA** (DF) - FONE: 225-1456 - **CURITIBA** (PR) - FONE: 33-6525 - **PORTO ALEGRE** (RS) - FONE: 42-3311 - **RIO DE JANEIRO** (RJ) - FONE: 252-8274 - **SÃO PAULO** (SP) - FONE: 283-3722.

REDE DE REVENDEDORES: BAURU (SP) - SOMA - 24-2558 - **BELÉM** (PA) - PAM - 222-9772 - **BELO HORIZONTE** (MG) - PROLOG - 337-5766 - **CAMPINAS** (SP) - MICROSYSTEMS - 51-3542 - **CAMPO GRANDE** (MS) - NEW LINE - 624-5349 - **CURITIBA** (PR) - DREHER - 224-0743 - **DUQUE DE CAXIAS** (RJ) - CPM - 771-0312 - **FORTALEZA** (CE) - GENERAL DATA - 226-2610 - **GOIÂNIA** (GO) - GENDADOS - 224-5486 / 224-5487 - **ITAJAÍ** (SC) - ENTEC - 44-0244 - **JUIZ DE FORA** (MG) - VERMAQ - 212-3809 / 213-5145 - **MANAUS** (AM) - CPD - 237-1033 / 237-1793 - **PIRACICABA** (SP) - SOGEMEC - 34-2100 - **RECIFE** (PE) - GENERAL DATA - 222-0357 / 222-2630 - **RESENDE** (RJ) - MICRO MAXI - 54-1644 / 54-2031 - **RIBEIRÃO PRETO** (SP) - KEHDI - 634-4715 - **RIO DE JANEIRO** (RJ) - EMBRASE - 222-4890 / 224-4379 - **SALVADOR** (BA) - SISMAQ - 243-0991 / 241-2894 - **SANTOS** (SP) - SISPER - 37-4705 - **SÃO JOSÉ DOS CAMPOS** (SP) - SISCON - 23-3752 - **SÃO LUIS** (MA) - M.P.A. - 221-1755 - **SÃO PAULO** (SP) - DATALOG - 211-9202 - **UBERLÂNDIA** (MG) - SILOGICA - 234-4191 - **VOLTA REDONDA** (RJ) - BEVORELLI - 42-2175 - SISCON - 42-1412.

calendário

CURSOS REGIONAIS/SCI

A SCI – Sistemas, Computação e Informática está implementando sua linha de Cursos Regionais, começando pelo Nordeste:

- Metodologia de Análise de Dados e Projeto de Banco de Dados – 09/08 a 12/08.
- Técnicas para o Desenvolvimento de Sistemas On-line – 29/08 a 01/09.

Procure o representante da SEI no Nordeste para maiores informações: BP Assessoria Técnica Ltda – Rua Monte Castelo, 87 – Tel.: (081) 221-0275 e 231-6124 – Recife.

DIGITAÇÃO E BASIC EM BH

A Minas Digital está promovendo cursos de **Digitação** e **Programação Basic.**

Para maiores informações: Rua Tupinambá, 1045 – Conjs. 601/2 – Belo Horizonte – Tel.: (031) 201-7555.

CURSOS/SYS INFORMATION

A Sys Information – Estudos e Assessoria Empresarial S/C Ltda promove para julho os seguintes cursos:

- Levantamento e Documentação de Sistemas – 13/07 a 16/07.
- Supervisão em Organização & Métodos – 27/07 a 30/07.

Ambos os cursos têm como instrutor Elazier Barbosa e a taxa de inscrição é de Cr$ 109.000,00. Está incluído no preço material de apoio dos cursos.

Informações à rua Eugênio de Medeiros, 278 – Pinheiros – São Paulo – Tel.: (011) 210-5292 e 813-4726.

BASIC/NASAJON

Está sendo promovido pela Nasajon Sistemas o curso de **Linguagem Basic** com duração de 20 horas, no horário das 19 às 21 horas.

O curso é ministrado em turmas de 10 alunos, com apostilas programadas. As aulas práticas são feitas em microcomputadores Digitus e Dismac. Preço do curso: Cr$ 25.000,00.

Informações à Av. Rio Branco, 45 Gr. 1311 – Rio de Janeiro – Tel.: (011) 263-1241.

I CONAI/SEMINÁRIO NACIONAL DE ROBÓTICA E AUTOMAÇÃO

Com a finalidade de tornar o I CONAI – Congresso Nacional de Automação Industrial e o II Simpósio Nacional de Robótica e Automação mais representativos para o setor, os organizadores de ambos – SEI, Sucesu-SP, Abicomp e Fórum das Américas – decidiram incorporar o Simpósio ao I CONAI. Os eventos acontecerão entre os dias 11 e 15 de julho no Hotel Maksoud Plaza. O último dia do evento será totalmente dedicado à programação do II Simpósio Nacional de Robótica e Automação. Os interessados podem obter maiores informações na sede da Sucesu-SP à rua Tabapuã, 627 – 1º andar – Tel.: (011) 852-2144 – São Paulo.

SUCESU-PR/CURSO

A Sucesu-PR, dando continuidade a sua programação para 1983, tem programado para 04 e 05 de julho o curso **O Impacto do Micro no CPD de Grande Porte.** O curso, com duração de 16 horas/aula, propõe-se a abordar a evolução da tecnologia em microcomputadores a nível internacional, o reflexo a nível brasileiro e a sua introdução nos CPD's de grande porte.

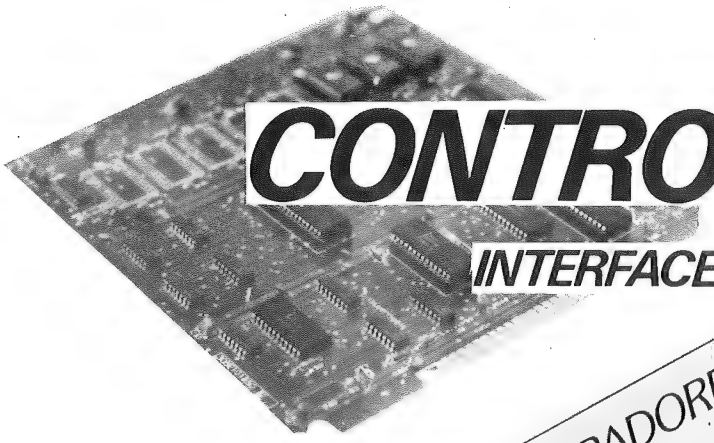
Maiores informações podem ser obtidas pelo telefone (041) 222-7613, com a Sr.ta Irene.

EASYTRIEVE/PANSOPHIC

A Pansophic realizará mais um curso Easytrieve – **Sistema de Recuperação de Informações e Geração de Relatórios** – no Rio de Janeiro, nos dias 05, 06 e 07 de julho.

Para maiores informações contate os escritórios da Pansophic:

RJ (021) 288-9495
SP (011) 284-0679
289-3740



CONTROLADORES

INTERFACES

2ª Parte

PROJETO DE SEPARADORES DE DADO E CLOCK PARA ACIONADORES DE DISCOS FLEXÍVEIS

Por Manuel Lois Anido

O Problema

Antes de apresentarmos os conceitos de PLO, descreveremos o problema que deve ser resolvido.

O problema consiste em se obter um sinal periódico gerado externamente ao acionador de disco que mantenha com o sinal lido do último uma diferença de fase conhecida, ou então, dentro de uma faixa previsível.

O PLO deverá ser imune aos eventuais ruídos de alta freqüência que possam ocorrer no sinal lido, e também deverá acompanhar, dentro de certos limites, as variações de baixa freqüência de tal sinal. O sinal periódico assim obtido será usado para auxiliar na separação do dado e **clock** do disco.

Há muitos exemplos que ilustram situações semelhantes, como por exemplo: para gerar uma base de tempo que acompanhe o sinal recebido e seja imune a ruídos atmosféricos ou outros tipos; em televisão — para sincronizar o oscilador horizontal ou regenerar a subportadora de croma etc.

Os separadores construídos com contadores ou monoestáveis, apresentados na primeira parte deste artigo, são muito mais sensíveis ao ruído, além de não se adaptarem às variações de freqüência do sinal de entrada. Por estes motivos eles não são utilizados em sistemas onde se exige alto desempenho.

Tipicamente, um separador com contadores ou monoestáveis apresenta até um erro de 10^8 (10 elevado a 8) **bits** lidos, enquanto um separador utilizando um PLO apresenta até um erro em 10^9 (10 elevado a 9), sendo o último, obviamente, muito superior.

A Solução

A solução freqüentemente utilizada para implementar um circuito que possua tais características é o PLO (Phase Locked Oscillator): um oscilador amarrado em fase), um tipo de servomecanismo cujas características descreveremos a seguir.

O PLO é um sistema eletrônico com realimentação, conforme ilustra a figura 1, e consiste basicamente de:

- **um comparador de fase**
- **um filtro passa-baixas**
- **um oscilador controlado a tensão (VCO)**
- **um contador divisor módulo *N***

Na figura 1 podemos ainda notar os seguintes elementos associados ao sistema:

f_i — Freqüência de entrada (Hz)
$K\phi$ — Ganho de conversão do comparador de fase (V/rad)
Kf — Ganho de conversão do filtro
Kv — Ganho de conversão do VCO (rad/s/v)
f0 — Freqüência de saída do PLO (HZ)

$$Kn = \frac{1}{N}$$

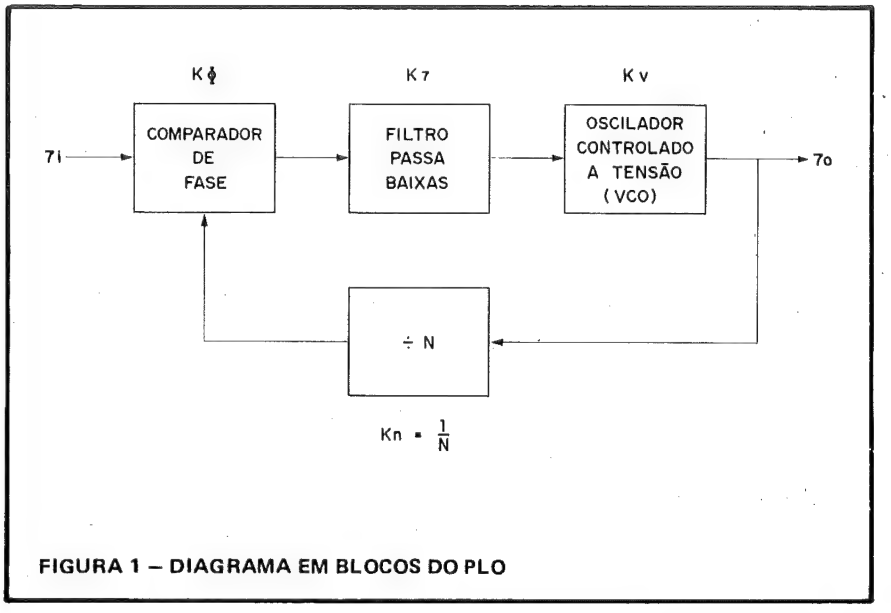


FIGURA 1 — DIAGRAMA EM BLOCOS DO PLO

Um sistema de controle em malha fechada pode ser representado da seguinte forma:

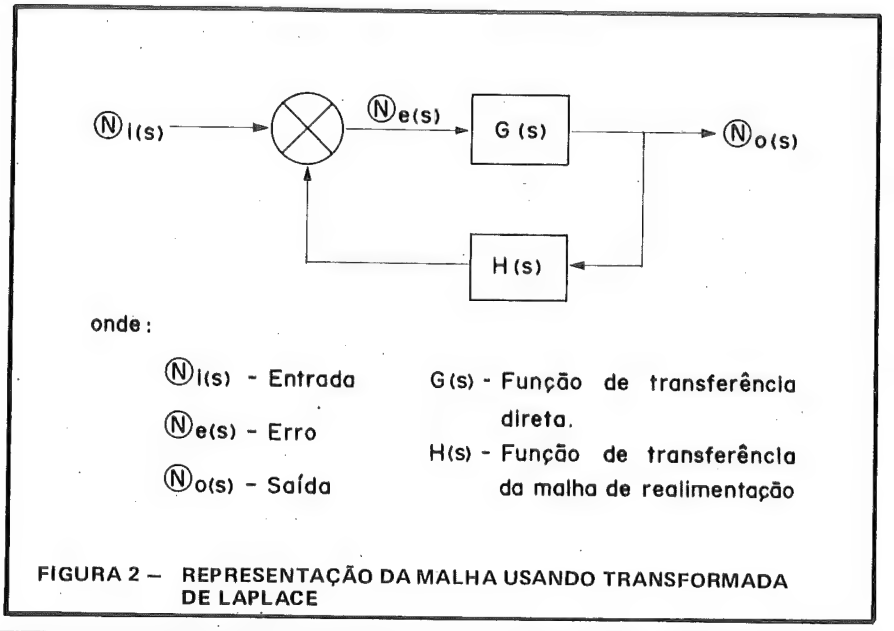


FIGURA 2 – REPRESENTAÇÃO DA MALHA USANDO TRANSFORMADA DE LAPLACE

$$\text{Tem-se que : } \Theta_0(s) = \frac{G(s)}{1 + G(s)\,H(s)}$$

O VCO (**Voltage Controlled Oscillator**) é um oscilador livre, cuja freqüência é normalmente determinada por uma malha com resistor-capacitor ou indutor-capacitor, A freqüência do VCO (f_0) é realimentada ao comparador de fase, que a compara ao sinal de entrada (f_i). Na saída do comparador de fase aparece uma "tensão de erro" que possui um nível médio DC proporcional à diferença ($f_i - f_0$) e fase $\Delta\,\phi$ entre a entrada e o sinal do VCO.

Esta tensão de erro é então filtrada, removendo assim interferências de ruídos de freqüências superiores a utilizada. Uma vez filtrada, esta tensão é aplicada ao VCO para completar a malha. A tensão de erro força a freqüência do VCO a mudar numa direção que reduz a diferença de freqüência entre a entrada e o VCO. Diz-se que a malha está na "faixa de captura" quando ele começa a mudar de freqüência para acompanhar o sinal de entrada. O processo continua até que as freqüências do VCO e de entrada sejam iguais. Neste ponto, a malha está sincronizada ou "amarrada em fase" e a freqüência do VCO é idêntica a da entrada (não considerando o divisor por N), **exceto por uma diferença de fase finita,** a qual é necessária para gerar a tensão de erro que desloca a freqüência do VCO, mantendo a malha amarrada em fase.

A ação contínua do PLO faz com que ele possa acompanhar as mudanças do sinal de entrada, quando estiver "amarrada em fase".

Pode-se dizer que o PLO possui três (3) estados distintos:

- **oscilação livre**
- **em captura**
- **amarrado-em-fase**

A faixa de freqüências na qual o sistema em malha seguirá mudanças na freqüência de entrada é chamada de "faixa de amarração" (**lock range**). A faixa de freqüências na qual a malha pode entrar em fase é a "faixa de captura" (**capture range**), a qual é sempre menor que a "faixa de amarração".

As características dinâmicas do PLO são ditadas pelo filtro passa-baixas. Se a diferença entre a freqüência de entrada e a do VCO for muito grande, o sinal de saída do comparador de fase pode ser muito grande para ser propagado através do filtro. Conseqüentemente, o sinal estará fora da faixa de captura da malha. Uma vez a malha sincronizada em fase, o filtro limita somente a velocidade com que ela pode acompanhar mudanças na freqüência de entrada. Além disso, o filtro fornece uma capacidade de memorização, assegurando uma recaptura rápida do sinal, caso o sistema saia de sincronismo, causado por um transiente de ruído.

Para compreender as equações e os conceitos de filtros que utilizaremos no decorrer deste trabalho, torna-se necessário que o leitor estude a bibliografia fornecida. Outra opção é a de simplesmente utilizar o circuito apresentado.

O projeto do separador PLO pode ser dividido em duas etapas, que são:

- **projeto do filtro passa-baixas**
- **circuito de recuperação do relógio**

PROJETO DO FILTRO PASSA-BAIXAS

O problema consiste em identificar o tipo de filtro adequado e o cálculo dos elementos do mesmo.

Considerando que a freqüência do sinal proveniente do acionador de disco estará sempre dentro de um certo intervalo, não possuindo aceleração constante e contínua, se escolhermos um sistema do tipo:

$$G(s)\,H(s) = \frac{K\,(s + a)}{s^2} \quad \text{teremos:}$$

erro de fase zero para entrada em degrau (ou seja, se o sinal do disco variar bruscamente de uma freqüência para outra, o PLO o perseguirá e após um certo tempo manterá com ele uma diferença de fase praticamente zero).

erro de fase zero para variações na entrada com velocidade constante (o PLO acompanha, mesmo que a freqüência de entrada varie com velocidade constante).

erro de fase constante para entrada com aceleração.

Do manual MOTOROLA, indicado na bibliografia, tem-se que:

$$R1.C = \frac{0{,}5\,K\phi Kv}{N\,Wn^2} \quad (I) \text{ e}$$

$$\frac{R2}{R1} = \frac{2\,\zeta\,Wn\,N}{K0\,Kv} \quad (II)$$

Onde $K\phi$, Kv e N são determinados a partir dos parâmetros dos circuitos empregados. Wn é escolhida de acordo com o fator de amortecimento desejado (ζ), a partir das curvas do filtro de 2ª ordem (Manual MOTOROLA).

DETERMINAÇÃO DE Wn

Para determinar Wn é preciso analisar o sistema disquete e especialmente a interação do controlador (INTEL-8272, NEC-μPD765 ou WD1793) com o separador PLO.

A sintetização de um filtro desse tipo pode ser feita com a configuração da figura 3.

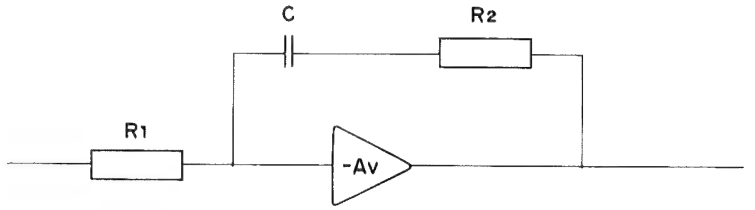


FIGURA 3 – FILTRO PASSA-BAIXAS QUE SATISFAZ ÀS EXIGÊNCIAS DO PROJETO

O PLO pode ser liberado em qualquer ponto da trilha, pois a cabeça atraca em qualquer lugar. Quando este é liberado, numa operação de leitura de identificação, há tempo suficiente para entrar em sincronismo. O pior caso, no entanto, ocorre quando já foi lida a identificação e o PLO só é liberado cinco (5) **bytes** antes do início do campo de dados, ver figura 4.

tem a metade da duração de um em densidade simples, o tempo é o mesmo e, conseqüentemente, o mesmo filtro passa-baixas pode ser usado nos dois casos.

Estipulando que a freqüência de saída do VCO deverá ter um erro menor que 5%, para um fator de amortecimento $\zeta = 0,7$ tem-se das curvas do manual MOTOROLA (pág. 25):

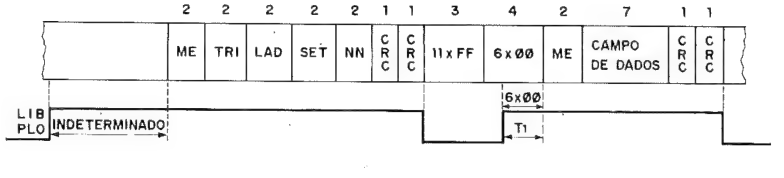


FIGURA 4 – INSTANTES DE LIBERAÇÃO DO PLO

Escolheremos como o tempo para o PLO entrar em sincronismo (**lock-up time**) um tempo correspondente a 4 **bytes,** ou seja, antes da marca especial (ME) o PLO já estará sincronizado.

Logo, este tempo é de:

$$4\ bytes \times \frac{8\ bytes}{byte} \times \frac{4\ \mu s}{bit} = 128\ \mu s$$

É importante observar que em **densidade dupla** o tempo T1 é o mesmo, pois a liberação do PLO ocorre 10 bytes antes do início do campo de dados. Como cada byte em densidade dupla

DETERMINAÇÃO DE KΦ

Para comparador de fase utilizaremos o C. I. MOTOROLA MC 4044, de cujo manual obtemos (pág. 24):

$$K0 = \frac{(2,25 - 0,75)\ V}{4\ \pi\ rad} = 0,12V/rad$$

Para obter uma faixa de captura quase simétrica foi utilizado um diodo de Germânio ($V_D = 0,2V$) em série com o amplificador a transistor existente no C. I. MC4044, polarizando a base em 1,4V.

DETERMINAÇÃO DE KV

A constante KV está amarrada à freqüência de oscilação do VCO, que no nosso caso será de 4 MHZ. O VCO utilizado é o C. I. MOTOROLA MC4024, e para ele foram levantadas diversas curvas com o auxílio de um capacitor variável. Verificou-se que $V_{in} = 3,8V$ oferecia um valor de tensão de entrada, em torno do qual se obtinha uma boa faixa linear na curva freqüência x tensão, conforme figura 5.

A faixa linear vai de 1,7 MHZ a 6,5 MHZ e corresponde a uma tensão de entrada de 3,2V a 4,8V, respectivamente.

$$Kv = \frac{6,5MHZ - 1,7MHZ}{4,8V - 3,2V} \times 2\ \pi\ rad/volt/s$$

$$\therefore Kv = 18,8 \times 10^6\ rad/s/volt.$$

DETERMINAÇÃO DE N

N corresponde à relação entre a freqüência do VCO (f_0) e a freqüência de entrada, que é de 250KHZ (numa seqüência de zeros).

$$N = \frac{4 \times 10^6\ HZ}{0,25 \times 10^6} = 16$$

$$Wnt = 4,5 \rightarrow Wn = \frac{4,5}{128 \times 10^{-6}} = 3,51 \times 10^4\ rad/s$$

O fator de amortecimento foi escolhido como $\zeta = 0,7$, pois a resposta do sistema não precisa ser excessivamente rápida, em detrimento do **ringing** apresentado, pois há tempo suficiente para a estabilização.

Agrupando, tem-se:

$$Wn = 3,51 \times 10^4\ rad/s$$
$$K0 = 0,12V/rad$$
$$KV = 18,8 \times 10^6\ rad/s/volt$$
$$N = 16$$

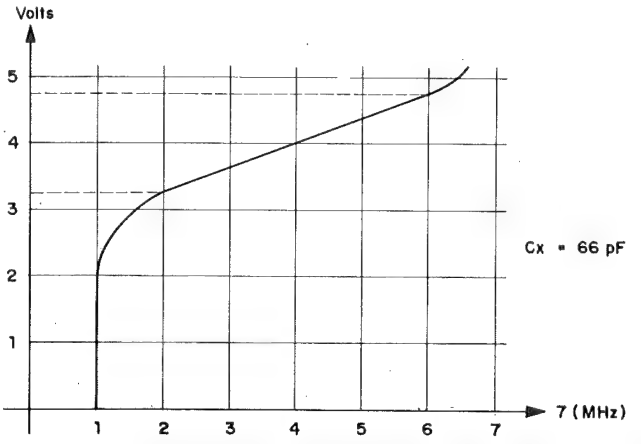


FIGURA 5 — CURVA FREQÜÊNCIA X TENSÃO DO VCO, PARA Cx = 66 pF

Determinação dos componentes do filtro:

$$R1C = \frac{0,5K\phi WKv}{NWn^2} = \frac{0,5 \times 0,12 \times 18,8 \times 10^6}{16 \times (3,51)^2 \times 10^8} = 5,7 \times 10^{-5}$$

$$R1C = 5,7 \times 10^{-5}$$

$$\frac{R2}{R1} = \frac{2\zeta Wn N}{K\phi Kv} = \frac{2 \times 0,7 \times 3,51 \times 10^4 \times 16}{0,12 \times 18,8 \times 10^6} \cong 0,35$$

$$\frac{R2}{R1} \cong 0,35$$

$$\text{Seja } R2 = 1K5\,\Omega \Rightarrow R1 = \frac{1,5 \times 10^3}{0,35} = 4,4K\ (2K2 + 2K2)$$

$$C = \frac{5,7 \times 10^{-5}}{4,4 \times 10^3} \cong 12\ KpF$$

Para filtrar altas freqüências na saída do comparador de fase utiliza-se um capacitor Cs, tal que:

$$10\left(\frac{1}{2} R_1\right) Cs = R_2 C$$

$$Cs = \frac{R_2 C}{5R_1} = \frac{1,5 \times 10^3 \times 12 \times 10^{-9}\ pF}{5 \times 4,4 \times 10^3}$$

$Cs \cong 820pF$
$R_1 = 2K2 + 2K2 - 1/8_W - 5\%$
$R_2 = 1K5 - \quad 1/8_W - 5\%$
$C = 12KpF - DISCO/32V - 5\%$
$Cs = 820pF - DISCO\ OU\ PLATE - 5\%$

CIRCUITO DE RECUPERAÇÃO DO RELÓGIO

O circuito de recuperação do relógio deve ser capaz de gerar a janela de leitura (**data window**) exigida pelos controladores, tanto em densidade simples como em dupla. O diagrama de tempos corresponde ao circuito de recuperação do relógio e é apresentado na figura 6.

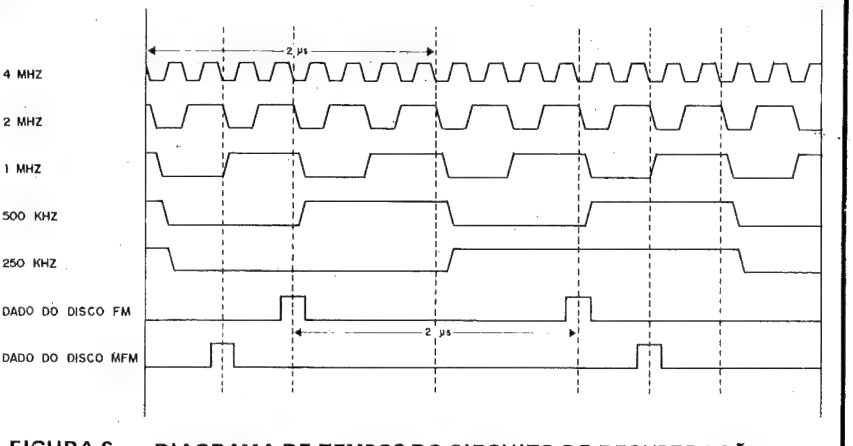


FIGURA 6 — DIAGRAMA DE TEMPOS DO CIRCUITO DE RECUPERAÇÃO DO RELÓGIO

A figura 7 apresenta o circuito final do PLO e nela podem ser vistos os contadores e multiplexadores que implementam o diagrama de tempos da figura 6.

OUTROS PLO'S

O separador PLO apresentado não é o mais compacto, mas é certamente um dos mais didáticos que pudemos encontrar.

Um separador compacto construído à base de contadores é o C. I. WD 9216 (**Western Digital**). Um outro C. I. para ser usado em PLO's é o WD1691, que segundo os testes realizados pelo autor apresenta um excelente desempenho.

A tendência atual é a de os controladores de disco flexível passarem a incorporar a lógica que era utilizada no PLO, como é o caso do C. I. WD 2797.

Os conceitos apresentados possuem larga aplicabilidade em áreas bem distintas e esperamos ter contribuído para auxiliar o leitor a resolver novos problemas.

Bibliografia:

- Design of Phase-Locked Loop Circuits — Howard M. Berlim — Howard W. Sams & Co., Inc.
- Intel 8272 — Single/Double — Density Floppy Disk controller.
- Scopus — Boletim nº 47 — Ano 4 — Maio/82.
- Signetics Analog Data Manual — Signetics Corporation.
- Phase Locked Loop Systems — Motorola Semiconductor Products Inc, 1973.
- Western Digital — WD1771/1793 Floppy Disk Controllers.
- Gardner, F. M. Phaselock Techniques. John Wiley & Sons, Inc., New York, 1966.
- Western Digital — Integrated Circuits Data Manual.

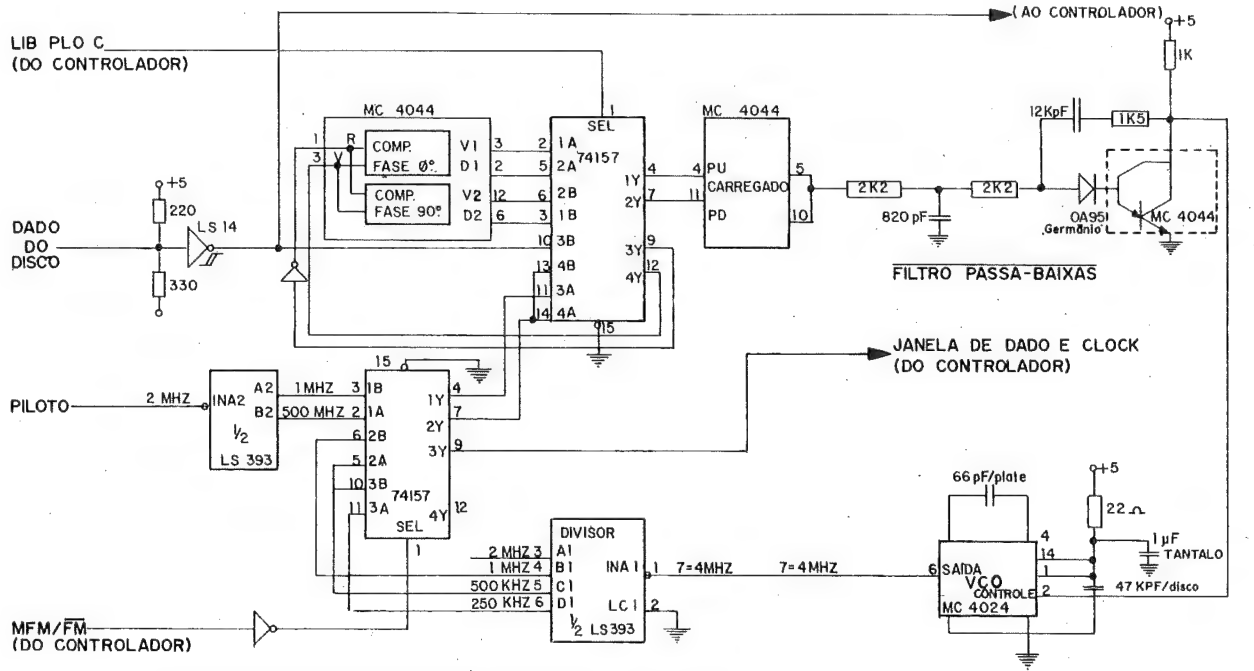


FIGURA 7 — ESQUEMA DE UM PLO PARA DISQUETE UTILIZANDO O COMPARADOR DE FASE MC4044 E O VCO MC4024

CONCURSO MICROCOMPUTADOR PESSOAL TK-82C

COMO PARTICIPAR

DO CONCURSO:

1 - Escreva uma monografia sobre o tema:
A Automação Bancária.
2 - As monografias deverão ser datilografadas em espaço dois, duas laudas.
3 - Os trabalhos deverão ser enviados até o dia 30/07 com os seguintes dados:
NOME, ENDEREÇO, PROFISSÃO.
4 - A promoção é aberta a pessoas de qualquer idade.
5 - O trabalho vencedor será publicado na revista INTERFACE nº 7 e seu autor ganhará um microcomputador pessoal TK-82C da MICRODIGITAL.
6 - Anexe à monografia uma autorização para cessão dos direitos de publicação do título.
7 - Mande seu trabalho para:
"INTERFACE - CONCURSO MICROCOMPUTADOR PESSOAL" - EST. DO TINDIBA, 2380 - RIO DE JANEIRO - RJ - CEP 22700.

O SISTEMA OPERACIONAL CP/M...

PARTE 5: O BDOS

Por Valdelírio P. S. Filho (Microsol Ind. Com. e Serv. Eletrônicos Ltda)

Dando continuidade a apresentação do BDOS (Basic Disk Operating System), passaremos, neste número, a descrever as funções relativas a disco (13 a 40).

Antes de prosseguirmos na descrição das funções executadas pelo BDOS, apresentaremos dois elementos importantes para o entendimento das explanações subseqüentes.

O LOGIN VECTOR

O BDOS mantém, internamente, um indicador de 16 bits chamado **LOGIN VECTOR**, destinado a indicar quais os drives que já foram selecionados desde que o equipamento foi ligado. No **LOGIN VECTOR** cada bit está associado a um drive (bit 0 – A, bit 1 – B,. . . ., bit 15 – P). Na primeira vez que um drive é selecionado, o BDOS "seta" o bit correspondente a este drive. Conforme poderemos constatar adiante, isto é feito, principalmente, para evitar que cada vez que um drive seja selecionado, o diretório do mesmo seja lido para efetuar a montagem do MAPA DE ALOCAÇÃO DE BLOCOS. Ao invés disto, ao selecionar um drive pela primeira vez, o BDOS monta na memória (no end. indicado no bloco de parâmetros de disco) o MAPA DE ALOCAÇÃO DE BLOCOS (ALV) correspondente ao drive selecionado e passa a utilizar-se do mesmo, sempre que necessita alocar ou liberar blocos neste drive.

O READ ONLY VECTOR

De forma semelhante ao **login vector**, o BDOS mantém um outro indicador de 16 bits, chamado **READ ONLY VECTOR**, destinado a indicar quais os drives que estão marcados como "só de leitura" (read only). Da mesma forma que no **login vector** cada bit corresponde a um drive (bit 0 – A, bit 1 – B, ..., bit 1 15 – P) e, estando "setado" indica que o drive associado não pode receber uma operação de escrita.

Já foi mencionado anteriormente, que todos os setores do diretório (ou uma parte destes) têm o seu **checksum** calculado a fim de detectar uma troca de disco (mídia). Pois bem, quando esta troca de disco é detectada, o bit do **READ ONLY VECTOR** correspondente ao drive no qual ocorreu a troca é "setado". Por outro lado, antes de efetuar uma escrita em qualquer drive, o BDOS verifica o estado do bit do **READ ONLY VECTOR** correspondente ao drive desejado. Se este estiver "resetado" a escrita é efetuada normalmente. Do contrário, o BDOS não escreve e exibe a mensagem:

Bdos Err On d: R/O

Tando o **READ ONLY VECTOR** como o **LOGIN VECTOR** são zerados quando o sistema é inicializado ou sofre uma partida a quente (WARMBOOT) e podem ser acessados por programa através de funções que serão mostradas adiante.

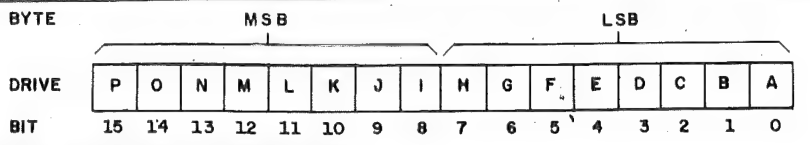


FIGURA 1 – FORMATO DOS INDICADORES LOGIN VECTOR E READY ONLY VECTOR

Prosseguiremos agora a descrição das funções executadas pelo BDOS.

13 – REINICIALIZA O SISTEMA DE DISCOS

Esta função é utilizada para, através de programa, levar o sistema de arquivos ao estado inicial de quando a máquina foi ligada. Para efetuar isto, o BDOS zera todos os bits do **login vector** e do **read only vector**, permitindo assim que todos os discos possam ser escritos (read/write), seleciona o drive A e assume o endereço de **default** para o buffer de disco (DMA ADDRESS = 0080H).

Não é aconselhável chamar esta função com algum arquivo em aberto pois, se o mesmo sofreu alguma escrita durante o período em que esteve aberto, os dados gravados serão perdidos, já que o BDOS só atualiza o diretório quando o arquivo é fechado.

Esta função é normalmente utilizada para permitir a troca de discos (mídia) durante a execução de um programa, sem que o drive correspondente seja marcado como **READ/ONLY**.

14 – SELECIONA DISCO

A presente função designa o drive de disco cujo número se encontra no registrador **E** (E = 0 para A, E = 1 para B, ... E = 15 para P) para ser utilizado como **default** nas próximas operações de disco. Ao executar esta função, o BDOS verifica inicialmente se o drive especificado já está logicamente selecionado e caso esteja, retorna ao programa chamador. Do contrário, chama SELDSK no BIOS e, de posse do endereço do BLOCO DE PARÂMETROS DE DISCO (DPH/DPB) do drive selecionado, move o conteúdo deste para uma área interna (ao BDOS) a fim de facilitar o acesso a estes dados. Feito isto, o BDOS verifica o estado do bit correspondente ao drive selecionado no **LOGIN VECTOR** e, se o mesmo estiver "setado", retorna de imediato ao programa chamador. Do contrário, "seta" este bit e lê todas as entradas do diretório a fim de montar a TABELA DE ALOCAÇÃO DE BLOCOS (ALV), "setando" os bits da tabela correspondentes aos blocos alocados pelas entradas.

Nas próximas operações com arquivos em que o campo "dr" do FCB for igual a 0 o BDOS assumirá o drive indicado através desta função. No entanto, para "dr" entre 1 e 16, será utilizado o drive especificado (A = 1, B = 2, ... P = 16).

15 – ABRE ARQUIVO

A função "abre arquivo" tem por objetivo ativar um arquivo que já existe no diretório, cujo FCB é endereçado pelo par DE. Para executar esta função, o BDOS substitui, temporariamente, o conteúdo do campo "dr" do FCB pelo número do usuário ativo, a fim de que só os arquivos pertencentes a este sejam considerados. Feito isto, o BDOS passa a comparar os bytes de 0 a 14 (μs a s2) do FCB com os bytes correspondentes de cada entrada do diretório, até encontrar uma que "case". Durante esta pesquisa, a presença de um "?" (ASCII 3FH) em qualquer um dos bytes de 1 a 14 do FCB (vide referência ambígua na parte 1) faz com que este byte seja irrelevante na comparação.

Ao encontrar uma entrada de diretório que "case" com o FCB, o BDOS copia o conteúdo das posições "d0" a "d15" (N^{os} dos blocos alocados) da entrada para as mesmas posições no FCB E "seta" o bit 7 do campo "s2" que é "resetado" quando o arquivo é escrito. Feito isto, o controle é devolvido ao programa chamador com o registrador A contendo um valor de 0 a 3 (posição relativa da entrada no buffer de diretório), se a abertura teve sucesso, ou 0FFH, se o arquivo não pode ser encontrado.

É importante observar que o ponto onde o arquivo será aberto é determinado pelo conteúdo dos campos "ex" e "s2", que determinam a extensão a ser acessada, e do campo "cr", que indica a posição do registro na extensão. Normalmente, quando o arquivo vai ser acessado seqüencialmente, estes campos são zerados a fim de que o arquivo seja aberto no primeiro registro. Quando o arquivo vai ser acessado aleatoriamente, é necessário que estes campos sejam zerados. O campo s1 é automaticamente zerado pelo BDOS.

16 – FECHA ARQUIVO

A função "fecha arquivo" executa o inverso da função "abre arquivo". Ela tem por objetivo gravar no diretório do disco as informações relativas à alocação de espaço contidas no FCB endereçado por DE, o qual, admite-se, corresponde a um arquivo previamente aberto. Da mesma forma que para a função "abre arquivo", o BDOS "varre" o diretório do disco especificado no FCB em busca de uma entrada que "case" com o FCB. Ao encontrar, o BDOS move o conteúdo dos campos "d0" a "d15" do FCB para as posições correspondentes da entrada de diretório.

Antes de executar as operações acima, o BDOS verifica o bit 7 do campo "s2" do FCB, a fim de determinar se o arquivo foi escrito durante o tempo em que esteve aberto. Caso este não tenha sido escrito (bit s^2 (7) = 1), o controle é devolvido, de imediato, ao programa chamador. Do contrário, as operações já mencionadas são executadas. Isto é feito com o intuito de minimizar o acesso ao diretório.

Ao retornar, o registrador A contém um valor entre 0 e 3 (posição relativa da entrada no buffer de diretório) caso o fechamento tenha sido efetuado com sucesso ou, 0FFH, se o arquivo não foi encontrado no diretório.

17 – BUSCA A PRIMEIRA

Esta função "varre" o diretório, desde o início, em busca de uma entrada que "case" com o FCB endereçado pelo par DE. Caso o arquivo desejado seja encontrado, o **buffer** de disco (DMA ADDRESS) é preenchido com o setor do

diretório que contém a entrada que coincidiu, e o registrador A retorna com um valor de 0 a 3, indicando a posição relativa da entrada no **buffer** de disco. Do contrário, ou seja, se o arquivo não for encontrado, o registrador A retorna com 0FFH.

Já que cada entrada ocupa 32 bytes, para calcular o endereço absoluto da entrada no **buffer** de disco basta multiplicar o conteúdo de A por 32 (ADD A 5 vezes) e adicionar o resultado ao endereço inicial do **buffer** (DMA ADDRESS).

Da mesma forma que na função **"abre arquivo"**, a presença de um "?" (ASCII 3FH) em qualquer um dos bytes de 1 a 14 do FCB faz com que este byte seja irrelevante na comparação. A presença de um "?" no campo "dr" (byte 0) do FCB, faz com que o drive de **default** seja utilizado e o campo "us" (nº do usuário que criou o arquivo) das entradas do diretório seja eliminado da comparação.

Esta função, normalmente, não é utilizada por programas aplicativos. A título de exemplo, podemos citar o utilitário STAT, que se utiliza desta função para calcular, a partir das informações contidas no diretório, o tamanho de um arquivo.

18 – BUSCA A PRÓXIMA

A função "busca a próxima" é similar à anterior, exceto que a pesquisa começa a partir da última entrada acessada pelo BDOS. Geralmente esta função é utilizada quando, após ter-se encontrado uma entrada de um dado arquivo, utilizando-se a função 17 (BUSCA A PRIMEIRA), deseja-se verificar se há uma outra entrada deste mesmo arquivo.

19 – APAGAR ARQUIVO

Esta função retira do diretório as entradas correspondentes ao(s) arquivo(s) especificado(s) no FCB endereçado por DE. Para executar esta função, o BDOS "varre" o diretório em busca de entradas que "casem" com o FCB e, ao encontrá-las, põe no campo "us" destas entradas o valor E5H. Além disto, o BDOS libera, no MAPA DE ALOCAÇÃO DE BLOCOS (ALV) do drive especificado, os blocos anteriormente alocados por esta entrada.

A presença, agora, do valor E5H no campo "us" da entrada passa a indicar que esta está livre, podendo ser utilizada para outros arquivos. A escolha do valor E5H deve-se ao fato de que, no padrão IBM de formatação de discos flexíveis, os setores são inicializados com este valor. Isto simplifica bastante a formatação de discos, evitando a necessidade de uma formação inicial do diretório.

Como podemos observar, tanto a(s) entrada(s) no diretório como os dados contidos no arquivo que foi "deletado" permanecem intactos no disco. Esta peculiaridade é utilizada por programas que recuperam arquivos apagados acidentalmente como, por exemplo, o utilitário RECDIR.COM do DGT-101 da DIGITUS. De forma simplificada, estes programas agem da seguinte forma:

> *Acham no diretório a(s) entrada(s) correspondente(s) ao arquivo desejado.*
>
> *De posse dos números dos blocos alocados pelo arquivo (campo "d0" a "d15" da(s) entrada(s)), verificam, no MAPA DE ALOCAÇÃO DE BLOCOS, se estes já foram alocados por outros arquivos.*
>
> *Se algum dos blocos já foi realocado, o arquivo se encontra, pelo menos parcialmente, destruído e o operador é informado. Do contrário, estes blocos são novamente alocados e a(s) entrada(s) reativada(s) substituindo-se o valor E5H do campo "us" pelo número do usuário corrente.*

Os campos de nome e tipo (bytes 1 a 11) do FCB podem conter uma referência ambígua (presença de "?"), possibilitando o apagamento de diversos arquivos de uma só vez. No entanto, o campo "dr" deve, obrigatoriamente, especificar o drive a ser utilizado (0 =**default**).

Em caso de sucesso ao retornar, o registrador A contém um valor entre 0 e 3 (posição relativa da entrada no setor de diretório). Caso o(s) arquivo(s) não tenha(m) sido encontrado(s), A retorna com FFH.

20 – LÊ SEQÜENCIALMENTE

Assumindo-se que o FCB endereçado por DE refere-se a um arquivo previamente aberto (pelas funções 15 a 22), esta função lê, do arquivo para o **buffer de disco,** (DMA ADDRESS) o registro (128 bytes) especificado pelo campo "cr". Se após o incremento, "cr" "estoura" (=128), a próxima extensão deste arquivo (se houver) é automaticamente aberta e o campo "cr" é zerado, habilitando o acesso ao próximo registro do arquivo.

Note que, neste caso, na próxima operação de leitura, se o programa chamador não alterou o conteúdo de "cr" (há restrições para esta alteração), o registro imediatamente seguinte será lido, fazendo-se assim um acesso estritamente seqüencial.

Em caso de sucesso, ao retornar o registrador A contém 00H, enquanto FFH é retornado quando chega-se ao fim (físico) do arquivo.

No próximo número concluiremos a apresentação das funções executadas pelo BDOS (2.2).

ERRATA

Na parte 3 (Nº 5, pág. 33), na descrição do campo s2 da entrada do diretório, onde está escrito "(arquivo de Mbytes)", leia-se "(arquivo de 8 Mbytes)".

Ainda na parte 3 (Nº 5, pág. 35), foi omitida a descrição do campo "0FF" do bloco de parâmetros de disco (DPB). Aqui vai a descrição:

0FF – Número de trilhas iniciais reservadas para sistema (COLD START + CCP + BDOS + BIOS). Normalmente são reservadas 2 trilhas.

ADVANCING
CONSULTORIA DE PESSOAL
Seleção de Pessoal com exclusividade na area de informática

ADVANCING
TREINAMENTO
Experiência na formação e aprimoramento de profissionais em P.D.

ADVANCING
COMPUTER SHOP
• Microcomputadores • Suprimentos
• Livros Técnicos • Software
• Manuais e Revistas •

ADVANCING-PRODUTOS E SERVIÇOS EM INFORMÁTICA LTDA. Andradas, 1580 – cj 518 – Fone 26.8248 e 26.0194 – P. Alegre – RS

CURSO Z-80

7ª Lição

Por André Gil Rubens e Ney Acyr de Oliveira
(A & N — Consultoria em Hardware e Software de Microcomputadores)

Nas lições anteriores estudamos o hardware do Z-80 através do seu funcionamento, da sua ligação com módulos de memória e entrada/saída e do seu sistema de atendimento de interrupções. Agora, começaremos a estudar o *software do Z-80*. Nesta lição, faremos uma revisão da aritmética complemento a dois (2's), números na representação BCD, e estudaremos a relação dos bits de condição do Z-80 com as operações lógicas e aritméticas. Em seguida, conheceremos os fundamentos e a sintaxe da linguagem ASSEMBLY do Z-80.

OS BITS DE CONDIÇÃO E AS OPERAÇÕES ARITMÉTICAS

Na segunda lição do nosso curso estudamos de forma sucinta os registros F F'. Agora, estudaremos a relação dos bits de condição com as operações aritméticas e desvios condicionais.

Trataremos este assunto com detalhes porque o poder de decisão de um programa, que é a sua "inteligência", é baseado nos testes dos bits de condição.

A disposição dos bits de condição nos registros F e F' está mostrada na figura 1.

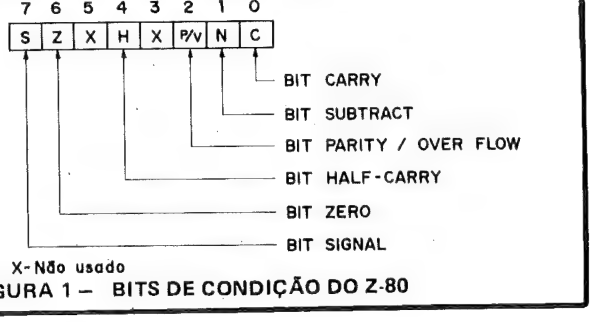
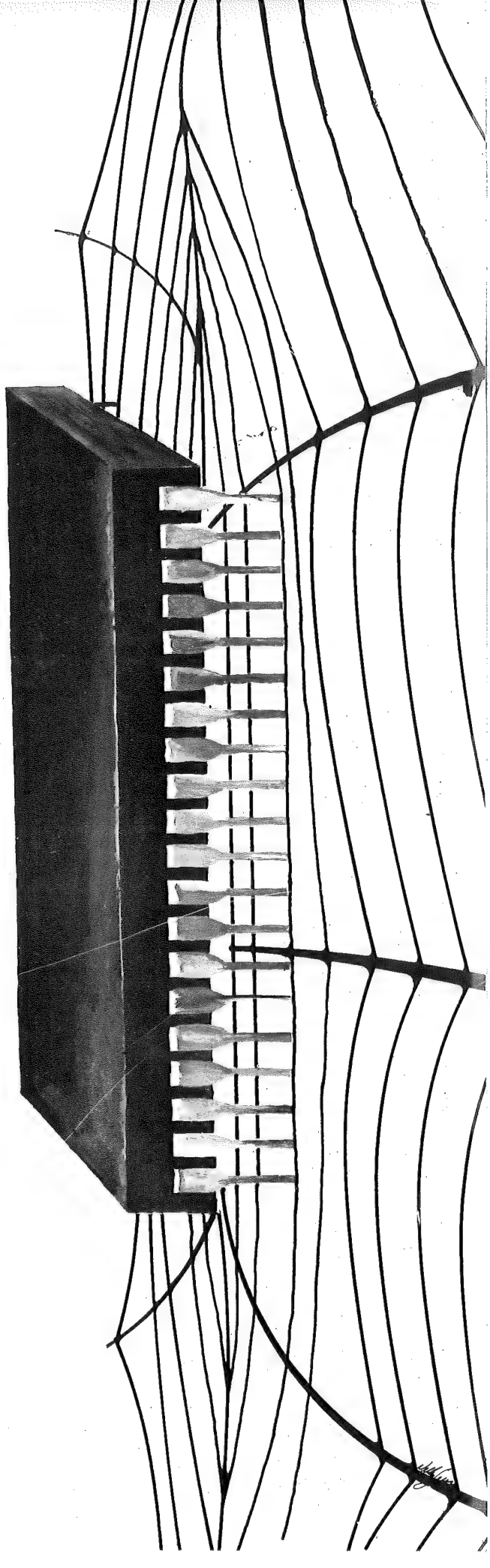


FIGURA 1 — BITS DE CONDIÇÃO DO Z-80

Estes bits de condição, que possuem significado próprio, estão agrupados logicamente no registro F ou F' de modo a facilitar o salvamento destes na pilha. Isto ocorre, principalmente, na chamada e retorno de sub-rotina e no atendimento de interrupções.

Os bits de condição C, P/V, S e Z podem ser testados pelas instruções que ocasionam desvios condicionais — JUMP's, CALL's e RETURN's — que na verdade avaliam os resultados de operações lógicas e aritméticas.



Os bits de condição H e N são usados para facilitar as operações aritméticas com números representados na notação BCD – Binary Coded Decimal.

O microprocessador 8080 não dispõe do bit N e o bit P/V tem a função de indicar, exclusivamente, a ocorrência de paridade, sendo identificado por **P**.

Visando simplificar as representações das várias bases numéricas no transcurso do texto usaremos a seguinte convenção: números terminados por "B" estão representados na base binária, por "D" na base decimal e por "H" na base hexadecimal.

REPRESENTAÇÃO COMPLEMENTO A DOIS

O hardware da Unidade Lógica e Aritmética (ULA) do Z-80 foi projetado para operar números algébricos na representação **complemento a dois (2's)**. A figura 2 mostra a forma geral desta técnica de representar números algébricos.

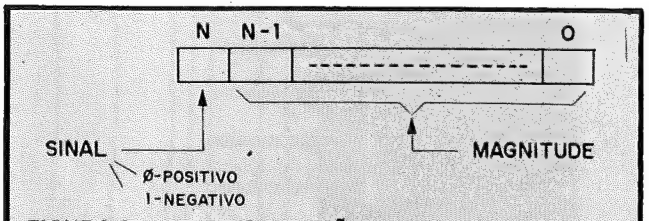


FIGURA 2 – REPRESENTAÇÃO 2's

O bit mais significativo representa o sinal do número; "0" representa o sinal positivo e "1" o negativo. O restante dos bits representam a magnitude do número.

Quando um byte representa um número na notação 2's, o bit 7 representa o sinal e os bits 0 a 6 representam a magnitude. A **magnitude de um número positivo** é a sua representação na base binária. Portanto, a faixa de números positivos que pode ser representada com **um byte** é de 0D a 127D, conforme ilustra a figura 3.

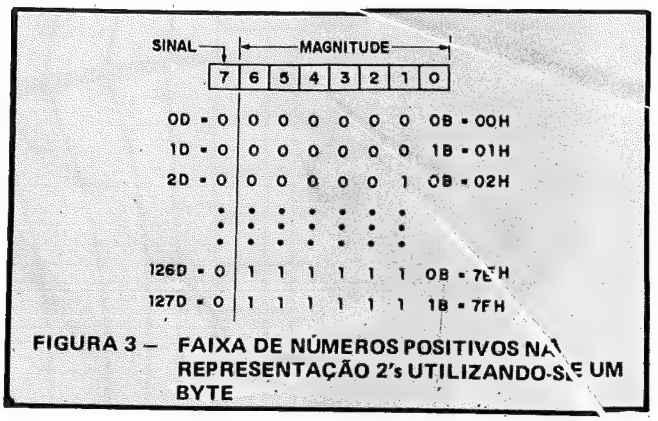


FIGURA 3 – FAIXA DE NÚMEROS POSITIVOS NA REPRESENTAÇÃO 2's UTILIZANDO-SE UM BYTE

A **mudança de sinal** de um número representado na notação 2's se faz da seguinte forma:

1. Cada bit é individualmente complementado, produzindo o chamado **complemento a 1 (1's).**
2. Somar "1" ao resultado anterior, desprezando, caso exista, o "vai um" do bit de mais alta ordem.

Por exemplo, – 10D representado na notação 2's é 11110110B, conforme ilustra a figura 4. Observe que o bit 7, representativo do sinal, ficou automaticamente setado, indicando um número negativo.

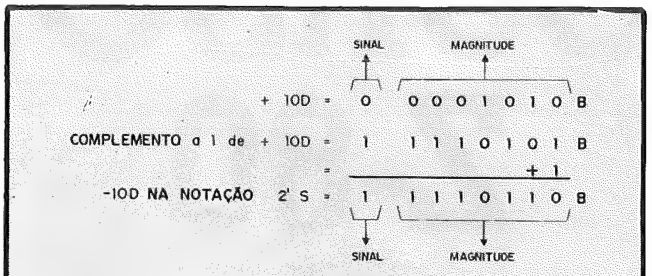


FIGURA 4 – CÁLCULO DO COMPLEMENTO 2's DE 10D

Portanto, a **faixa de números negativos** que pode ser representada com um **byte** é de -1_{10} a -128_{10}, conforme ilustra a figura 5.

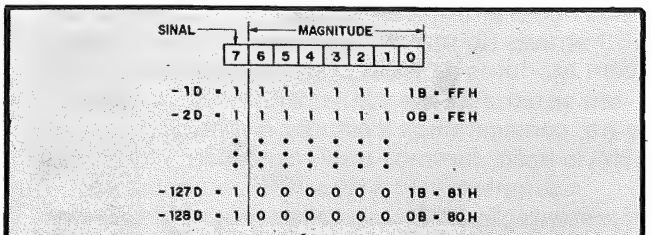


FIGURA 5 – FAIXA DE NÚMEROS NEGATIVOS NA REPRESENTAÇÃO 2's, REPRESENTADOS POR UM BYTE

As operações de subtração com números representados na notação 2's são realizados **somando-se** o minuendo ao subtraendo de sinal invertido (2's), e desprezando-se o possível "vai um".

Por exemplo, a instrução SUB E subtrai o conteúdo do **registro E** do conteúdo do **acumulador** e armazena o resultado no próprio acumulador. Esta operação realiza-se conforme ilustra a figura 6. Vamos assumir que em **A** temos 1AH e em **E** temos 0CH. Então, mudamos o sinal de 0CH e somamos a 1AH, obtendo como resultado 0EH, ignorando-se o "vai um"

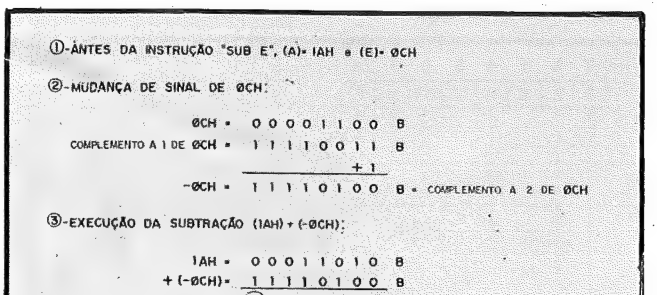
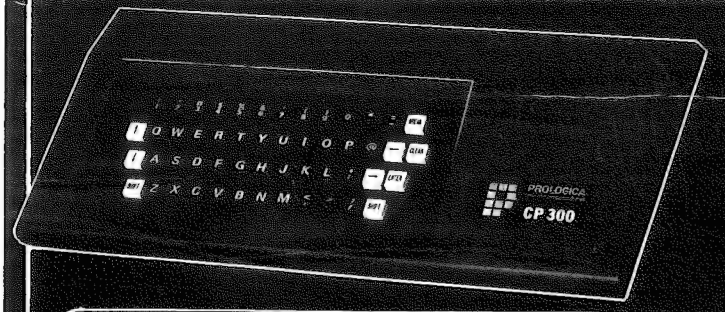


FIGURA 6 – EXECUÇÃO DA INSTRUÇÃO SUB E

Note que para realizar subtrações de números representados na notação 2's basta que o hardware tenha circuitos inversores e o mesmo somador usado nas operações de adição. Isto simplifica bastante a implementação da unidade lógica e aritmética, razão pela qual a notação 2's é usada na maioria dos computadores digitais.

O **bit de condição S** indica o sinal do resultado e, nada mais é que o bit mais significativo (bit 7).



O MICRO QUE JÁ NASCEU GRANDE

CP-300

CP-500 · CP-300 · CP-200

CARACTERÍSTICAS TÉCNICAS

- CPU com microprocessador Z80 de 2MHz memória principal de 48KB.
- Teclado alfanumérico com 54 teclas e função auto repeat-reset com proteção contra acionamento involuntário.
- Conexões – para TV ou monitor de vídeo – 16 linhas com 64 col. – 16 linhas com 32 col. – modo gráfico com 48 x 128 pontos.
- Interface para gravador com velocidade de 500 e 1500 Bauds.
- Expansões para: controlador de drives, porta paralela (impressora) e porta serial RS 232 (Modem, etc.).
- Dimensões: altura: 7,5 cm – compr.: 33,0 cm – larg.: 22,0 cm – peso: 1,5 kg.

Compatível em todos os aspectos com o CP-500, inclusive no software, porém é composto de módulos que podem ser adquiridos de acordo com suas necessidades.

PAGAMENTOS SEM ACRESCIMO

PROMOÇÃO ESPECIAL

SOFTWARE

JOGOS:

BATALHA AÉREA
- CP-200 (F) . 3.850,00
- CP-500 (F/D) 4.400,00*

BATALHA NAVAL
- CP-200 (F) . 5.830,00

FORCA
- CP-200 (F) . 4.400,00
- CP-500 (F/D) 4.620,00*

TIRO AO ALVO
- CP-200 (F) . 3.850,00
- CP-500 (F/D) 4.400,00*

BIORRITMO
- CP-200 (F) . 3.850,00

LOTO
- CP-200 (F) . 4.180,00

TABUADA
- CP-200 (F) . 3.850,00
- CP-500 (F/D) 3.080,00*

SIMULADOR DE VÔO
- CP-200 (F) . 6.600,00

COMANDO UFO
- CP-200 (F) . 3.850,00
- CP-500 (F/D) 4.950,00*

OESTE SELVAGEM
- CP-200 (F) . 4.620,00

SENHA
- CP-200 (F) . 4.950,00

INVASÃO CÓSMICA
- CP-200 (F) . 6.050,00

SÔMENTE P/CP-500

PATRULHA (F/D) 6.160,00*

INVASORES (F/D) 5.610,00*

PADDLE PINBALL (F)
Simula jogo de fliperama 9.600,00

DISCOS VOADORES (F)
Controle o canhão para abatê-los 8.000,00

DANCING DEMOND (F/D) - Incrível demoninho dançarino . . 8.000,00*

XADREZ (F/D)
6 níveis que vão desafiá-lo 10.000,00*

CUBO (F/D)
Você nunca resolveu o cubo? A solução é cubo 6.400,00*

JOGOS EM BASIC (F/D) 6.400,00*

JOGOS (F/D) COSMIC, SCARFMAN, LUNAR, BARRICADA, GALAXI, METEOR, PENETREITOR - cada . 9.600,00*

PROMOÇÕES

PACOTE ECONÔMICO (F)
Para CP-200 NE-Z8000, com música, damas, gamão, conta corrente e controle de estoque 4.000,00

10 JOGOS EM DISCO
Para CP-500, variedade: boa, sky, pouso lunar, jornada, teaser, cupim, hopper, cram, fireman, spacefire . . . 24.000,00

APLICATIVOS

SOMENTE P/CP-200 (F)

CONTAS A PAGAR
Controla o acumulado do mês, ano e operações e poupança . . 13.200,00

AGENDA
Trabalha com dois tipos de informação atividade e telefone 8.800,00

CADASTRO DE CLIENTE
Cadastra p/cliente a razão social, rua, bairro, cid., est., CEP., tel., produto adquirido, data da aquisição 13.750,00

VÍDEO-TÍTULO
Cria títulos para vídeo-tapes, inclusive com movimentos 16.500,00

VU-CALC
Destina-se à execução de cálculos sobre uma planilha 10.000,00

SOMENTE P/CP-500

CADASTRO DE CLIENTES (D) . . . 20 ORTN

MALA-DIRETA (D/I)
Sistema de mala direta permitindo a impressão de etiquetas utilizando ou não chave de acesso 40.000,00

FINANÇAS (D)
Engloba funções de juros compostos, análise de preço de venda e rendimentos, taxas de retorno, tabela de amortização, saldo hipotecário, pagamentos a prazo . 40.000,00

PROCALC (D)
Destina-se a execução de cálculos sobre uma planilha 120.000,00

VÍDEO (F/D)
Editor gráfico da tela. Desenhe com facilidade no CP-500. Armazena as telas em fita ou diskette 10.000,00*

BANNER (F/D/I)
Imprime mensagens em letras garrafais (80 col.) 6.400,00*

SCRIPY (I)
Completo e versátil processador de texto . . . . . em fita . . . . 32.000,00 em disco . . . 40.000,00

CONTROLE DE AÇÕES (D/F)
Para quem gosta da Bolsa. Mostra os resultados (Lucro/Perda) 6.400,00*

DIRETÓRIO (D)
Organiza e cadastra todos os seus programas e arquivos em diskette automaticamente. Pesquisa por disco ou programa e pode imprimir ordenadamente 16.000,00

BANCO DE DADOS (D/I)
Sistema de fichário eletrônico. Você mesmo cria as fichas e pode pesquisar de vários modos, além de permitir impressão. Similar ao Profile 40.000,00

CARTA ASTRAL (F/I)
Este programa faz todos os cálculos necessários ao desenho de uma carta astral e ainda imprime a mesma em questão de minutos. Só não interpreta 15.000,00

UTILITÁRIOS (SOMENTE P/CP-500)

CONVERT (F/D)
Converte números decimais e hexa. Pode ser chamado a qualquer instante e não atrapalha o BASIC 6.400,00*

LISTA (D/I)
Imprime as listagens de programas em BASIC de forma limpa e organizada. Não se perca com LLIST 16.000,00

SUPERTECLA (F)
Cada tecla de seu CP-500 representa duas palavras do BASIC além de seu valor normal. Reduz drasticamente o tempo gasto na teclagem de programas 8.000,00

ODONTO (F/D)
Engloba setor financeiro, agenda, cadastro, mala direta, lay-out dentário . . . . . 50 ORTN

EDITOR (D)
Poderoso editor assembler para os que programam em linguagem de máquina. Manual c/instruções 32.000,00

SOUND (F/D)
Pequena mas útil subrotina que cria uma nova palavra em BASIC – "SOUND". Instruções detalhadas de como colocá-la em seus programas 6.400,00*

Kapram

F - Para CP-500/CP-300 ou CP-200 em fita. I - Requer impressora.
D - Para CP-500/CP-300 em disco. * - Acrescentar Cr$ 6.000,00 p/versão em disco

filcres

Filcres Importação e Representações Ltda.
Rua Aurora, 165 – CEP 01209 – São Paulo – SP
Telex 1131298 FILG BR – PBX 223-7388 – Ramais 2, 4, 12, 18, 19 – Diretos: 223-1446, 222-3458, 220-5794 e 220-9113 - Reembolso - R17 Direto: 222-0016, 220-7718

NOME . . . . . TEL. . . . . .
END. . . . . CEP INT8
CID. . . . . EST. . . . . EQUIP. . . . . .
ENVIE AS INFORMAÇÕES ACIMA PARA **FILCRES IMP. E REPR. LTDA.**

O **bit de condição Z** é setado (Z = 1) se o resultado de uma operação for igual a zero. Caso contrário, assume o valor zero (Z = 0).

OVERFLOW

Nas operações de soma ou subtração de números de oito bits, representados na notação 2's, os resultados são armazenados no acumulador, que é um registro de 8 bits.

Se o resultado de uma soma de dois números positivos ultrapassar 127D, ou se o resultado de uma soma de dois números negativos for inferior a - 128D, significa que tais resultados não podem ser representados somente pelos 8 bits do acumulador. Esta condição é chamada de **overflow**, isto é, o resultado da operação não coube no registro a ele destinado.

O **bit de condição P/V** indica o overflow em operações aritméticas. Se ocorrer tal condição após uma operação aritmética o bit P/V é setado (P/V = 1). Assim, ele nos alerta para a validade ou não do resultado.

Na figura 7 temos dois exemplos de ocorrência de overflow. No primeiro caso, temos a soma de dois números positivos de 8 bits, cujo resultado ficou negativo. No segundo caso, temos a soma de dois números negativos, cujo resultado ficou positivo.

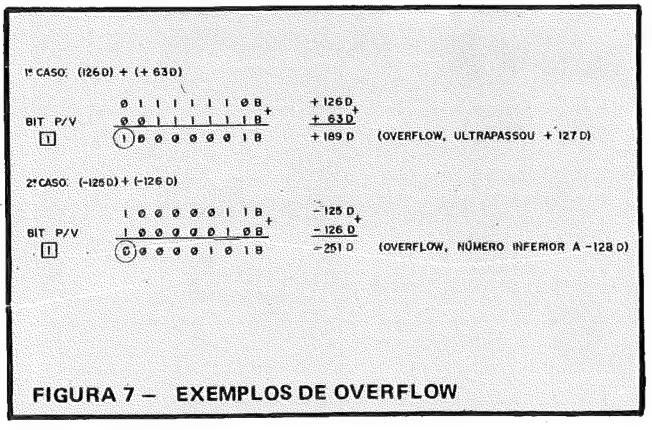


FIGURA 7 – EXEMPLOS DE OVERFLOW

OPERAÇÕES COM NÚMEROS 2's SEM O SINAL

Se um byte for interpretado como sendo a magnitude de um número binário positivo representado na notação 2's, poderemos estender a faixa de números positivos representados por um byte para 0D a 255D, conforme ilustra a figura 8.

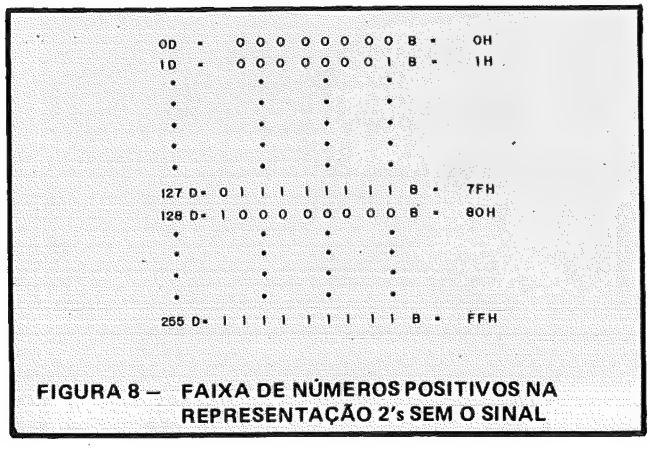


FIGURA 8 – FAIXA DE NÚMEROS POSITIVOS NA REPRESENTAÇÃO 2's SEM O SINAL

Note que as operações de soma e subtração na aritmética 2's continuam válidas.

Na realização de uma soma, se ocorrer "vai um" significa que o resultado ultrapassou 255_{10}, fazendo com que o bit de condição C fique setado (C = 1).

As instruções que realizam operações de subtração, automaticamente, invertem o "vai um" do resultado, antes de armazená-lo no bit de condição C. Tal inversão permite que o bit C indique o sinal do resultado de operações de subtração com números na representação 2's sem o sinal.

Assim, o bit C resetado (C = 0) significa que o resultado é maior ou igual a zero e, o bit C setado (C = 1) simboliza um resultado negativo.

A figura 9 exemplifica uma operação de subtração de números positivos representados na notação 2's sem o sinal, usando-se uma instrução de subtração. Para facilitar o entendimento do sinal foi adicionado um **nono bit fictício.** Observe que o bit C é o inverso do "vai um", o que significa que o resultado é positivo.

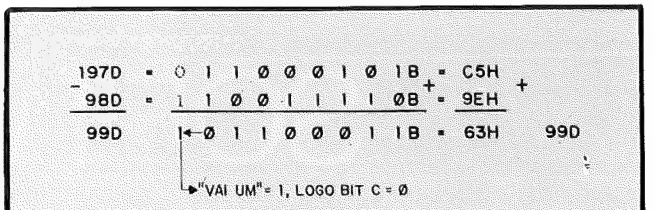


FIGURA 9 – OPERAÇÃO DE SUBTRAÇÃO DE DOIS NÚMEROS POSITIVOS NA REPRESENTAÇÃO 2's SEM SINAL

REPRESENTAÇÃO BCD

Na representação BCD (Binary Coded Decimal) cada dígito decimal é representado pelo seu correspondente código binário. A tabela da figura 10 mostra a representação binária de cada dígito decimal.

DECIMAL	BINÁRIO
0 D	0 0 0 0 B
1 D	0 0 0 1 B
2 D	0 0 1 0 B
3 D	0 0 1 1 B
4 D	0 1 0 0 B
5 D	0 1 0 1 B
6 D	0 1 1 0 B
7 D	0 1 1 1 B
8 D	1 0 0 0 B
9 D	1 0 0 1 B

FIGURA 10 – REPRESENTAÇÃO BCD

A figura 11 ilustra a representação dos números 89_{10} e 135_{10} na notação BCD.

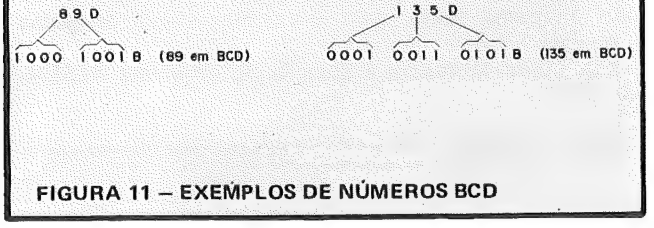


FIGURA 11 – EXEMPLOS DE NÚMEROS BCD

OS BITS DE CONDIÇÃO "H" E "N"

Os bits de condição H e N não podem ser testados pelas instruções de desvio e servem para auxiliar nas operações aritméticas com números na representação BCD.

O **bit H** setado (H = 1) simboliza a ocorrência de "vai um" do quarto para o quinto bit do resultado.

O **bit N** setado (N = 1) simboliza que a operação realizada foi de subtração e, caso contrário (N = 0), simboliza a realização de uma operação de adição.

Na realidade, a Unidade Lógica e Aritmética (ULA) do Z-80 foi projetada para operar números na representação 2's. No entanto, em muitas aplicações fica mais cômodo operar números em BCD, principalmente, quando envolve a manipulação de dados de entrada ou saída.

O Z-80 dispõe da instrução "DAA", que realiza os ajustes dos resultados obtidos quando números BCD são adicionados ou subtraídos pela ULA. Isto porque a ULA assume que está operando números representados na notação 2's. A figura 12 ilustra a função da instrução "DAA". Observe que os bits de condição H e N também são parâmetros de entrada desta instrução.

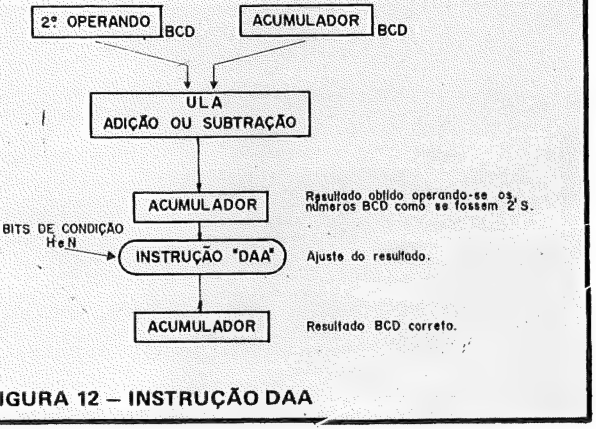


FIGURA 12 – INSTRUÇÃO DAA

Para o caso de operações de adição, isto é, se N = 0 o resultado é ajustado pela instrução "DAA" da seguinte maneira:

1. Se H = 1, subtrair seis do dígito BCD menos significativo.
2. Se C = 1, subtrair seis do dígito BCD mais significativo.
3. Se C = 1 e H = 1, subtrair seis de ambos os dígitos BCD.

A figura 13 mostra alguns exemplos de operações de adição com números BCD.

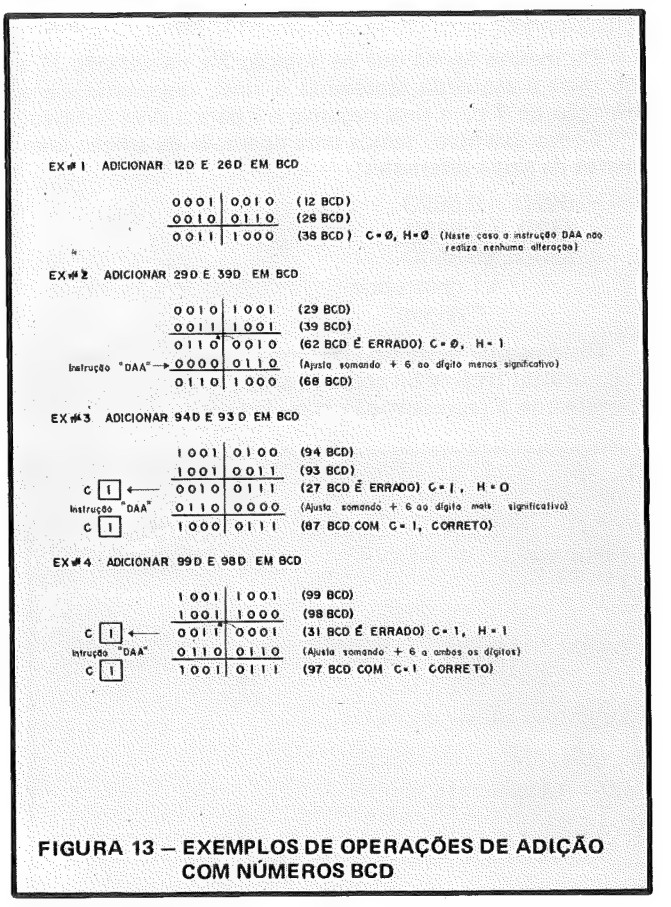


FIGURA 13 – EXEMPLOS DE OPERAÇÕES DE ADIÇÃO COM NÚMEROS BCD

Para o caso de operações de subtração, isto é, se N = 1 o resultado é ajustado pela instrução "DAA", subtraindo seis ao dígito BCD nas seguintes condições:

1. Se H = 1, somar seis ao dígito BCD menos significativo.
2. Se C = 1, somar seis ao dígito BCD mais significativo.
3. Somar seis ao dígito BCD que tiver valor superior a 1001B.

A figura 14 mostra alguns exemplos de operações de subtração com números BCD. Note que os resultados dos exemplos 3 e 4 são representados na notação complemento a 10 (10's), que é semelhante à forma binária 2's.

O 10's é obtido subtraindo-se cada dígito de nove e somando-se um ao dígito menos significativo do resultado anterior.

As operações de subtração com números BCD representados na notação 10's são realizadas somando-se o minuendo ao subtraendo, complementado a 10, e desprezando-se o possível "vai um".

Por exemplo, o 10's de 72 é 27 + 1 = 28, que corresponde ao resultado do exemplo 3 da figura 14.

O exemplo abaixo compara uma operação de subtração normal e utilizando a notação 10's:

subtração normal	subtração com 10's
91D	91D
− 72D	+ 28D
19D	19D
	1 (despreza o "vai um")

Da mesma forma que na notação 2's, na notação 10's o bit mais à esquerda representa o sinal, respeitando-se a convenção de 1 para sinal negativo e 0 para o sinal positivo.

Como exemplo, temos uma operação de adição com números de sinais diferentes:

```
+ 692D  = 0.692D
- 342D  = 1.658D
+ 350D    0.350D
          (1 despreza o
           "vai um"
```

Observe que neste tipo de representação é muito mais prático utilizar o carry bit como sinal do resultado, de maneira semelhante à representação complemento a 2 sem sinal.

```
EX#1 SUBTRAIR 10D DE 98D EM BCD

          1001 | 1000   (98 BCD)
        - 0001 | 0000   (10 BCD)
          1000 | 1000   (88 BCD) C=0, H=0 (NESTE CASO A INSTRUÇÃO
                                            DAA NAO REALIZA NENHUMA
                                            ALTERAÇÃO)

EX#2 SUBTRAIR 19D DE 91D EM BCD

          1001 | 0001   (91 BCD)
        - 0001 | 1001   (19 BCD)
          0111 | 1000   (78 BCD ERRADO) C=0, H=1
INSTRUÇÃO "DAA" 0000 | 0110   (AJUSTA SUBTRAINDO 6 DO DÍGITO MENOS SIGNIFICATIVO)
          0111 | 0010   (72 BCD COM C=0, CORRETO)

EX#3 SUBTRAIR 91D DE 19D EM BCD

          0001 | 1001   (19 BCD)
        - 1001 | 0001   (91 BCD)
C [1]     1000 | 1000   (88 BCD E ERRADO) C=1   H=0
          0110 | 0000   (AJUSTA SUBTRAINDO 6 DO DÍGITO BCD MAIS SIGNIFICATIVO)
C [0]     0010 | 1000   (28 BCD COM C=0, CORRETO)

EX#4 SUBTRAIR 99D DE 11D EM BCD

          0001 | 0001   (11 BCD)
        - 1001 | 1001   (99 BCD)
C [1]     0111 | 1000   (78 BCD ERRADO) C=1, H=1
          0110 | 0110   (AJUSTA SUBTRAINDO 6 DE AMBOS OS DÍGITOS)
C [0]     0001 | 0010   (12 BCD COM C=0, CORRETO)
```

FIGURA 14 – EXEMPLOS DE OPERAÇÕES DE SUBTRAÇÃO COM NÚMEROS BCD

LINGUAGEM ASSEMBLY

Tendo em vista as limitações inerentes à conversão manual das instruções de um programa, nos seus respectivos códigos binários, todos os fabricantes de microcomputadores oferecem um **PROGRAMA MONTADOR OU "ASSEMBLER".**

O montador tem a função de converter qualquer programa codificado na linguagem simbólica **ASSEMBLY,** chamado **programa fonte,** para o correspondente **programa objeto** em código de máquina.

A figura 15 ilustra a atuação do programa **ASSEMBLER.**

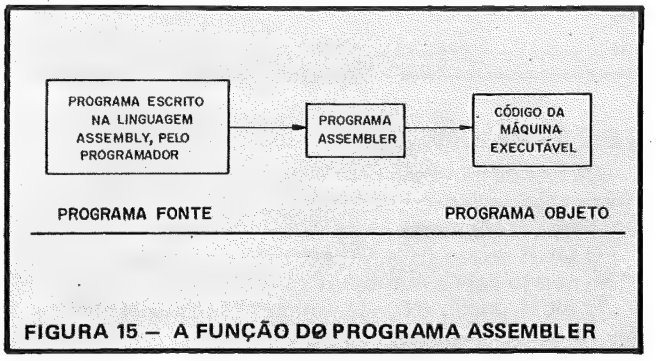


FIGURA 15 – A FUNÇÃO DO PROGRAMA ASSEMBLER

A linguagem ASSEMBLY além de possibilitar a representação das operações por **mnemônicos,** que é um conjunto de 2,3 ou 4 letras, permite o uso de endereços e operandos simbólicos. Para tanto, durante o processo de montagem de um programa, além da tarefa de conversão dos códigos das operações, o montador realiza os cálculos dos endereços e operandos simbólicos.

A figura 16 mostra um programa fonte escrito pelo programador na linguagem Assembly do Z-80, e o correspondente programa objeto em hexadecimal.

Na realidade, o Z-80 executa o programa objeto em código de máquina quando este encontra-se armazenado na memória.

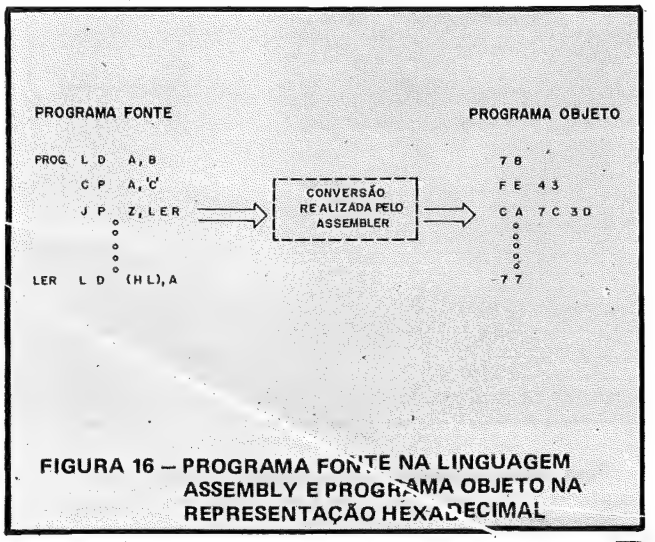


FIGURA 16 – PROGRAMA FONTE NA LINGUAGEM ASSEMBLY E PROGRAMA OBJETO NA REPRESENTAÇÃO HEXADECIMAL

A&N

Consultoria em hardware e software de microcomputadores

Projetos de Interfaces
Controle de processos
Teleprocessamento

DESENVOLVIMENTO DE SOFTWARE
Aplicações Comerciais
Aplicações em tempo real

André Gil Rubens – CREA 30911 – D
Ney Acyr R. Oliveira – CREA 38324 – D Tel.: (021) 248-8105

Clube do Microsoftware

Esta coluna publicará os programas que os leitores nos enviarem mensalmente. Os interessados em publicar seus programas deverão remetê-los para: REVISTA INTERFACE – MICROCOMPUTADORES. ESTRADA DO TINDIBA, 2380 – JACAREPAGUÁ – RJ – 22700

PROGRAMA 12: CADASTRO DE DISQUETES

Programa enviado pela Digitus – Ind. Com. Serv. Eletrônicos – Belo Horizonte – MG, para o microcomputador DGT-100.

Este programa foi desenvolvido visando um auxílio para os usuários de microcomputadores dotados de disc-drives e que possuam grandes quantidades de arquivos em disquetes. Ele permite que você cadastre cada uma de suas disquetes, exigindo para isso que você as numere (de 00 a 99).

Para você usar este programa basta que siga as instruções mandadas por ele. Lembre-se de que todo programa novo deve ser cadastrado usando o comando de recadastramento de disco. É um bom costume estabelecer um prazo para novo recadastramento de todos os seus discos, como por exemplo de mês em mês. Se você tiver uma impressora o programa permite que você liste seu arquivo de disco por programas ou por discos, além de listá-los no display.

PROGRAMA 12: CADASTRO DE DISQUETES

```
10  REM **NR é o número de records; N$ é o arranjo dos
    records e a primeira letra é o número da disquete:
    A$ ( ) é o arranjo que contém os records do disco que está
    sendo analisado no momento: IN é o número da
    disquete que está sendo cadastrada.
20  REM * * IN é o número da diskete que está sendo cadastrada:
    NR é o número de records já cadastrados.
30  CLEAR 24000
40  DIMA$ (61): DIMN$ (2000)
50  CLS: INPUT "Escolha uma opção:
    1 – Novo arquivo
    2 – Processador arquivo
    ";X
60  ONXGOTO70, 920
70  NR = - 1
80  CLS
90  INPUT "Qual o número que vai ser dado a esta
    disquete (2 dígitos)"; IN$
100 IFLEN (IN$) < > 2GOTO90
110 FORJ = 0TONR
120 IFLEF T$ (N$ (J), 2) = IN$ THENPRINT "Disco já existente.
    Use comando < 7 > para recadastrar": GOSUB830: GOTO930
130 NEXT
140 N = O: INPUT "DE QUAL DRIVE VOCÊ VAI CADASTRAR"; D ·
150 IFD = OTHENCMD "DIR $0": EN = 64 ELSE CMD "DIR 1": EN = 0
160 FORA = &H3C00 + & H80 + EN TO & H3FFF STEP & H40
170 PRINT @ 0, A
180 IFPEEK (A) = 32 OR CHR$ (PEEK (A)) = "?" GOTO240
190 A$ = " ": FORB = ATOA + 59
200 IFPEEK (B) = 32THEN GOSUB330 ELSE A$ = A$ + CHR$ (PEEK (B) )
210 NEXT
220 NEXT
230 PRINT "fim na 155?": END
240 A = 0
250 FORNR = NR + 1 TONR + N - 1
260 PRINT @ 0, NR
270 N$ (NR) = IN$ + A$ (A)
280 A = A + 1
290 NEXT
300 NR = NR - 1
310 GOTO930
320 END
330 IFA$ = "MOUNT" THEN240
340 A$ (N) = A$: N = N + 1: B = B + 2 + 12 – LEN (A$): A$ = " "
350 IFPEEK (B + 1) = 32THENB = A + 59
360 RETURN
370 OPEN "I", 1, "DIR/DAD"
380 INPUT # 1, NR
390 ' INPUT # 1, IN
400 FORJ = 0TONR
410 PRINT @ 0, J
420 INPUT # 1, N$ (J)
430 NEXT
440 CLOSE
450 RETURN
460 REM * * LISTAR NO PRINTER * *
465 J = 0
470 CLS: INPUT "Você quer listagem por ordem dos
    discos (1) ou dos programas (2)"; EN
480 IFEN < > 2 AND EN < > 1 THEN470
490 IFEN = 2THEN A$ = "ORDEM DOS PROGRAMAS": CMD "O", NR + 1,
    N$ (0) (3, 12) ELSE A$ = "ORDEM DOS DISCOS": CMD "O", NR + 1, N$ (0)
500 LPRINTCHR$ (14) "LISTAGEM POR"; A$
505 LPRINT
510 LPRINTCHR$ (27) "D" "CHR$ (2) CHR$ (15) CHR$ (25) CHR$ (40) CHR$ (50) CHR$ (65)
520 IFPEEK (14312) < > 63THEN520
530 POKE14312,0
540 LPRINTCHR$ (9) "PROGRAMA";
550 LPRINTCHR$ (9) "DISCO";
560 LPRINTCHR$ (9) "PROGRAMA";
570 LPRINTCHR$ (9) "DISCO";
580 LPRINTCHR$ (9) "PROGRAMA";
590 LPRINTCHR$ (9) "DISCO"
610 PG = 5
630 LPRINTCHR$ (9) RIGHT$ (N$ (J), LEN (N$ (J) ) - 2); CHR$ (9) LEFT$ (N$ (J), 2);
635 J = J + 1
640 LPRINTCHR$ (9) RIGHT$ (N$ (J), LEN (N$ (J) ) - 2); CHR$ (9) LEFT$ (N$ (J), 2);
642 J = J + 1
645 LPRINTCHR$ (9) RIGHT$ (N$ (J), LEN (N$ (J) ) - 2); CHR$ (9) LEFT$ (N$ (J), 2)
650 J = J + 1: PG = PG + 1
660 IFN$ (J) = " " THENGOTO930ELSEIFPG < 66THENGOTO630ELSELPRINTCHR$ (140): GOTO500
666 PRINT "LINHA666": END
670 REM * * Procurar um programa * *
680 CLS: INPUT "QUAL O PROGRAMA A SER PROCURADO", A$
690 FORJ = 0TONR
700 IFA$ = RIGHT$ (N$ (J), LEN (N$ (J) ) - THENPRINT "VEJA
    DISQUETE No. → "; LEFT$ (N$ (J), 2)
710 NEXT
720 GOSUB830
730 GOTO930
740 REM * * LISTAR NO VÍDEO * *
750 CLS: INPUT "Você quer listagem por ordem dos discos (1) ou
    dos programas (2)"; EN
760 IFEN < > 2 AND EN < > 1THEN470
770 PRINT @ 0, "Espere um momento
780 IFEN = 2THENCMD "O", NR + 1, N$ (0) (3, 12) ELSE CMD "O", NR + 1, N$ (0)
790 FORJ = 0TONR
800 PRINTRIGHT$ (N$ (J), LEN (N$ (J) ) - 2); "→ "; LEFT$ (N$ (J), 2)
810 NEXT
820 GOSUB830: GOTO930
830 PRINT: INPUT "APERTE RETURN QDO PRONTO PARA CONTINUAR"; RT
840 RETURN
850 REM * * LISTAR UMA DISQUETE * *
860 CLS: INPUT "Qual disco a ser listado (2 dígitos)" ; IN$
870 IFLEN (IN$) < > 2GOTO860
880 FORJ = 0TONR
890 IFLEFT$ (N$ (J), 2) = IN$ THENPRINTRIGHT$ (N$ (J),
    LEN (N$ (J) ) - 2); "→ "; LEFT$ (N$ (J), 2)
900 NEXT
910 GOSUB830: GOTO930
920 CLS: GOSUB370
930 CLS: PRINT" MENU
940 PRINT "1 – CADASTRAR NOVA DISQUETE 2 – LISTAR NO PRINTER"
950 PRINT "3 – LISTAR NO VÍDEO 4 – PROCURAR UM PROGRAMA
960 PRINT "5 – LISTAR UMA DISQUETE 6 – GUARDAR ARQUIVO
970 PRINT "7 – RECADASTRAR DISQUETE
980 INPUT "COMANDO"; X
990 ONXGOTO80, 460, 740, 670, 850, 1010, 1120
1000 GOTO930
1010 REM * * GUARDAR ARQUIVO * *
1020 CLS
1030 OPEN "O", 1, "DIR/DAD"
1040 PRINT # 1, NR
1050 ' PRINT # 1, IN
1060 FORJ = 0TONR
1070 PRINT @ 0, J
1080 PRINT # 1, N$ (J)
1090 NEXT
1100 CLOSE
1110 GOTO930
1120 REM * * Recadastrar uma disquete * *
1130 REM * * Primeiro detectar qtos records tem a disquete IN$.
    Este valor é QR * *
1140 REM * * Detectar início dos records da
    disquete IN$. Este valor é PT
1150 REM * * Chegar para cima todos records. Loop for next"
1160 INPUT "Qual o número da disquete a ser
    recadastrada (2 dígitos)"; IN$
1170 IFLEN (IN$) < > 2GOTO1160
1180 QR = 0
1190 CLS
1200 FORJ = 0TONR
1210 PRINT @ 0, J: PRINT @ 5, QR
1220 EFLEFT$ (N$ (J), 2) = INSTHENQR = QR + 1
1230 NEXT
1240 IFQR = 0THENPRINT "Não existe esta
    disquete": GOSUB830: GOTO 930
1250 FORJ = 0TONR
1260 IFLEFT$ (N$ (J), 2) = IN$ THENPT = J: J = NR
1270 NEXT
1280 FORPT = PTTONR
1290 N$ (PT) = N$ (PT + QR)
1300 NEXT
1310 NR = NR – QR
1320 GOTO140
```

ERRATA:

PROGRAMA 09 (Revista nº 6)

```
20   PRINT AT N, 0; "■"; AT N, 31; "■"
35   PRINT AT 21, N; "■"; AT 0, N; "■"
140  PRINT AT X, Y; "▣"
300  PRINT AT M, N; "✚"
```

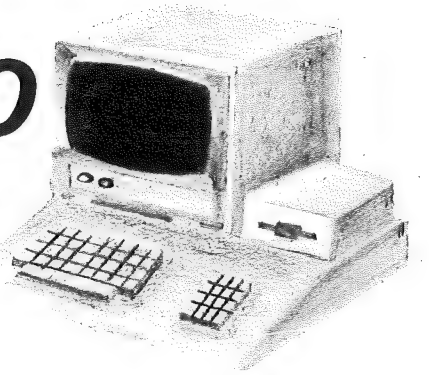
Obs:
SÍMBOLOS GRÁFICOS:

Linha 20 e 35
GRAPHICS + SPACE (ESPAÇO EM VÍDEO REVERSO)

Linha 140
GRAPHICS + SHIFT H

Linha 300
GRAPHICS + SHIFT B (ASTERÍSTICO EM VÍDEO REVERSO)

MICRO COMPUTADORES



ARQUIVO DE DADOS EM DISCO
VERSUS
ARQUIVO DE DADOS EM CASSETE

2ª Parte

Por Cesar da Costa

Na primeira parte, edição nº 7, vimos através de dois programinhas como os dados digitados no teclado são armazenados na memória e dali transferidos e gravados na fita cassete, automaticamente. Vimos também como os dados gravados na fita cassete são transferidos para a memória e dali impressos no vídeo. Nesta segunda parte do artigo, usando os mesmos programas anteriores e introduzindo algumas alterações, mostraremos como se deve fazer para que o microcomputador controle uma gravação e uma leitura de fita cassete, e como criar arquivos de dados e dimensioná-los numa fita.

Na primeira parte do artigo após introduzirmos os dois programas na máquina, digitarmos os 10 strings para valor de **A$** e verificarmos que os mesmos haviam sido gravados na fita cassete, observamos que a fita continuava avançando, ou seja, o microcomputador não reconhecia o fim do programa. O que fazer? Inicialmente dê um **RESET** no microcomputador para apagar de sua memória os programas anteriores; introduza uma modificação no programa **FITA CASSETE PARA MEMÓRIA PARA O VÍDEO,** linha 235, e entre com os dois programas, conforme se segue:

```
100  REM **TECLADO PARA MEMÓRIA PARA
     FITA CASSETE ***
110  CLS
120  INPUT A$
130  PRINT # - 1, A$
140  GOTO120
200  REM ***FITA CASSETE PARA MEMÓRIA
     PARA O VÍDEO***
210  CLS
220  INPUT # - 1, A$
230  PRINT A$
235  IF A$ = "HOJE É DIA DE FESTA" THEN END
240  GOTO 220
```

Retroceda a fita até o início, digite RUN 210, pressione RETURN e aperte PLAY no gravador; a fita avança e as informações surgem no vídeo:

```
A B C
1 2 3 4 5 6 7 8 9 0
AZUL
DOMINGO
LEVE PEPITA PARA A FESTA
TESTE
UM, DOIS, TRÊS
INTERFACE
MICROCOMPUTADOR
HOJE É DIA DE FESTA
READY
```

A palavra **READY** no vídeo, ao final da impressão dos 10 strings, indica que o microcomputador executou o programa com sucesso e a fita cassete no gravador está totalmente parada.

O string **HOJE É DIA DE FESTA** indica um "fim de arquivo", ou seja, o fim da informação na fita. Caso quiséssemos colocar mais informações nesta fita não haveria problema, bastando indicar na linha 235 o string que encerrava o seu arquivo de dados. Recomendamos ao leitor marcar sempre o início e o fim de cada programa, utilizando o contador de fita do gravador cassete.

Um grupo particular de dados de informações gravados numa fita cassete é conhecido como "registro". Um grupo de registros constituem um "arquivo".

Normalmente, um arquivo é constituído de dois ou mais programas; um programa monta o arquivo na fita cas-

Toda vez que se projeta um sistema de controle para um processo, é axiomático que a velocidade do sistema de controle seja superior a dos movimentos do processo controlado. Completando 10 anos de serviços à informática, a BK Controles Eletrônicos Ltda., empresa 100% nacional, com seus sistemas de suporte e fornecimento elétrico de alta confiabilidade, manteve o pioneirismo, no setor (mais de 10.000 usuários), criando as mais diversas soluções para instalações dos principais centros de processamento do país.

- Sistema BK Série V - estabilizador eletrônico de tensão AC (de 0,8 kva até 300 kva).
- Sistema No Break-NBK - Automação Bancária (de 150 va até 3.000 va)

Em lançamento:

- Sistema BK/Elene - Filtro regulador de linha (de 100 va até 3.000 va)

BK vende confiabilidade.

Tecnologia para o ano 2.000

SELLER

ANO 10
CONTROLES ELETRÔNICOS
100% NACIONAL

BK controles eletrônicos ltda.

ESCRITÓRIO CENTRAL E FÁBRICA - PORTO ALEGRE - RS
Av. João Ferreira Jardim, 138 - Bairro Rubem Berta
CEP 90000 - Fone: (0512) 40-3611 - Telex (051) 2303

FILIAL SÃO PAULO - SP
Av. Indianópolis, 2171 - CEP 04063 - Fones: (011) 275-5817 e 275-4510 - BIP (011) 815-3344 - BIP 30MJ Telex: (011) 37308

REPRESENTANTES NAS PRINCIPAIS CAPITAIS

sete e o outro o executa. A seguir apresentamos um arquivo que poderá ser útil para você: **Catálogo personalizado de telefones.**

```
100  REM ***FAÇA UM CATÁLOGO DE NOMES E
     NÚMEROS TELEFÔNICOS ***
110  CLEAR 100
120  CLS
130  PRINT: INPUT "NOMES"; NAYM$
140  INPUT "NÚMEROS"; NMBR$
150  PRINT # - 1, NAYM$, NMBR$
160  GOTO130
200  REM ***LEIA 10 NOMES E
     NÚMEROS ***
210  CLEAR 100
220  CLS
230  FOR K=1  TO 10
240  INPUT # - 1, NAYM$, NMBR$
250  PRINT NAYM$ TAB (30) NMBR$
260  IF NAYM$ = "NÃO HÁ MAIS NOMES"
     THEN END
270  NEXT K
300  REM ***DECLARAÇÃO DE CONFIRMAÇÃO
     DOS NOMES ***
310  PRINT
320  PRINT "PARA INSERIR MAIS NOMES
     E NÚMEROS",
330  PRINT "PRESSIONE A BARRA DE ESPAÇO";
340  KEY$ = INKEY$: IF KEY$ = " " THEN 340
350  IF KEY$ = " " THEN 220 ELSE 340
```

Não existe nada de novo nestes programas. Leia-os, imagine-os funcionando, e então execute-os. No vídeo surgirão as seguintes perguntas: NOME?, NÚMERO?.

Responda com os nomes e números desejados. Cada nova linha faz avançar uma linha para o topo do vídeo; isto não afeta a gravação da informação na fita cassete.

e pressione **RETURN.** Aperte **PLAY** no gravador e você verá o seu arquivo de nomes e números no vídeo.

```
INTERFACE          392-8965
PUBLICIDADE        240-5200
COSTA SERRA        397-4521
TAVARES CUNHA      395-5000
TORRES TROVÃO      385-3104
GIOVANA AIRES      375-4022
PAULO ANTONIO      315-3142
CYRO MONTES        325-1317
MAURO ANGUSTO      741-3022
PAULO ANTUNES      824-3112
```

Lembre-se que para inserir mais de 10 nomes é necessário pressionar a barra de espaço; assim sendo, no vídeo surge o grupo de nomes e números restantes.

Como não há mais nomes, o microcomputador identifica o fim do programa e "pára" automaticamente o gravador.

```
GUERRA FLORES      953-2142
ISIDORO CRAVO      900-0120
LUIZ CORTES        901-0430
NÃO HÁ MAIS NOMES  999 999 9999
READY
>
```

Até aqui, na fita cassete, você armazenou somente valores de strings como variável nas expressões PRINT # - 1. Por exemplo:

```
130  PRINT # - 1, A$ ◄—— strings ——┐
150  PRINT # - 1, NAYM$, NMBR$ ◄———┘
```

Podemos também armazenar números na fita cassete usando variáveis numéricas ou expressões. Por exemplo:

```
140  PRINT # - 1, X ◄—— variável numérica
```

```
NOME? INTERFACE
NÚMERO? 392-8965
NOME? PUBLICIDADE
NÚMERO? 240-5200
NOME? COSTA SERRA
NÚMERO? 397-4521
NOME? TAVARES CUNHA
NÚMERO? 395-5000
NOME? TORRES TROVÃO
NÚMERO? 385-3104
NOME? GIOVANA AIRES
NÚMERO? 375-4022
NOME? PAULO ANTONIO
NÚMERO? 315-3142
```

```
NOME? CYRO MONTES
NÚMERO? 325-1317
NOME? MAURO AUGUSTO
NÚMERO? 741-3022
NOME? PAULO ANTUNES
NÚMERO? 824-3112
NOME? GUERRA FLORES
NÚMERO? 953-2142
NOME? ISIDORO CRAVO
NÚMERO? 900-0120
NOME? LUIZ CORTES
NÚMERO? 901-0430
NOME? NÃO HÁ MAIS NOMES
NÚMERO? 999 999 9999
```

Arbitre o número 999 999 9999 para o nome **NÃO HÁ MAIS NOMES.**

Retroceda a fita cassete até o início, digite **RUN 210** e pressione **RETURN.** Aperte **PLAY** no gravador e você verá o seu arquivo de nomes e números no vídeo.

O programa a seguir armazena na fita cassete os números inteiros consecutivos, imprimindo cada número no vídeo antes de gravá-los na fita.

```
100  REM **MEMÓRIA PARA VÍDEO E FITA **
110  CLS
120  X = 1
130  PRINT X;
140  PRINT # - 1, X (memória para vídeo)
150  X = X + 1 (memória para fita)
160  GOTO 130
200  REM **FITA PARA MEMÓRIA PARA VÍDEO **
210  CLS
220  INPUT # - 1, X
230  PRINT X;
240  GOTO 220
```

Entre com ambos os programas no microcomputador; coloque uma fita cassete C-10 (5 minutos em cada lado) no gravador, pressione **PLAY** e **RECORD**, digite **RUN** e pressione **RETURN**. Surge no vídeo os números inteiros consecutivos; em cerca de 5 minutos a fita chega ao fim e o gravador "pára". Porém, o microcomputador continua listando os números no vídeo.

Dê um **RESET** no micro, retroceda a fita, digite **RUN 210** e pressione **RETURN** e **PLAY** no gravador.

Os números gravados na fita são lidos um a um, e surgem no vídeo:

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
67 68 69 70 71 72 73 74 75
READY
>
```

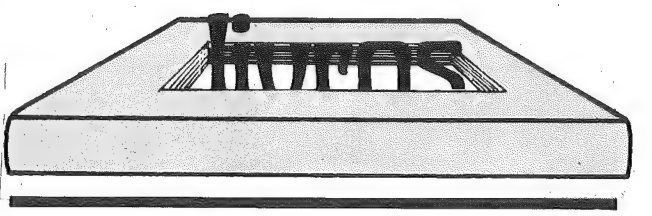
Conforme vemos, usando-se a expressão PRINT # - 1, X pode-se colocar uma média de 75 números em um dos lados de uma fita cassete C-10.

A tabela 1 apresenta a quantidade de números que podem ser inseridos em diversos tamanhos de fitas cassete, utilizando a expressão PRINT # - 1, X.

CASSETE	TEMPO POR LADO	QUANTIDADE DE NÚMEROS
C-30	15 minutos	225
C-60	30 minutos	450
C-90	45 minutos	675

Mais tarde veremos que escrever somente um número de cada vez é o caminho menos eficiente para armazenar dados numa fita cassete. Como podemos dobrar a capacidade de armazenamento de uma fita? No próximo número apresentaremos a solução. Até lá.

livros livros



PEREIRA FILHO, Jorge da Cunha — "BASIC PARA MICROS PESSOAIS", Editora Campus Ltda, Rio de Janeiro.

BASIC PARA MICROS PESSOAIS desenvolve a apresentação da linguagem BASIC de forma didática e gradual. Através de exemplos o leitor utilizando o seu microcomputador pessoal poderá testar a aquisição dos conhecimentos constantes do texto ao longo de seus 10 capítulos, que além desses exemplos para execução em computador, trazem também numerosos exercícios subdivididos em duas partes: um guia de estudo dirigido e problemas para serem desenvolvidos e testados em microcomputador.

O guia de estudo dirigido se destina a ressaltar os principais conceitos e conhecimentos que o leitor assimila com a leitura dos capítulos. Os problemas permitem uma efetiva aplicação dos conhecimentos adquiridos.

A linguagem é fácil e direta, e mesmo aqueles que não têm experiência anterior poderão escrever os seus principais programas em BASIC utilizando um microcomputador pessoal. Em particular, o livro detalha as qualidades da linguagem BASIC dos pequenos microcomputadores pessoais do tipo teclado plano e similares, como TK 82-C, TK-85, CP-200, NEZ-8000, SINCLAIR ZX 80, ZX 81, ZX SPECTRUN e TIMEX 1000.

Todavia, mesmo para as pessoas que possuam microcomputadores de maior porte, este livro é muito útil como introdução, já que estes apresentam em comum a maioria das facilidades dos pequenos micros.

MEMÓRIAS

1ª Parte

Por Cesar da Costa

O custo por bit de um computador é proporcional à sua capacidade de memória, ou seja, quanto maior a sua capacidade de memória mais alto é o seu custo. Como funcionam os dispositivos de memória? Sua evolução e suas características. Tipos e Tecnologias envolvidas em sua construção.
Nesta série de artigos apresentamos um histórico dos dispositivos de armazenamento, desde os primeiros computadores até os mais modernos, dando ênfase ao hardware envolvido em seus projetos.

O funcionamento de um computador pressupõe a presença de unidades de armazenamento das informações que devem ser tratadas. Nas operações aritméticas ou de controle, os dados são armazenados em registros de operações que constituem os elementos básicos de armazenamento. Os registros são constituídos por flip-flops onde cada flip-flop funciona como um bit de armazenamento, de modo que um registro com N flip-flops poderá armazenar N dados.

Há outra categoria de registros que armazenam dados, sob a forma de bits, que não vão ser imediatamente processados e que dispõem de uma codificação (endereço) para onde a informação é enviada (escrita) ou de onde é recolhida para processamento (leitura). Estes registros são conhecidos como "memórias".

O processo de transferência de dados (escrita ou leitura) a partir desses registros deve ser bastante rápido, mas haverá sempre um compromisso entre a velocidade e o tamanho do registro, de modo a não encarecer muito o custo por bit de memória, o que está sendo superado por novas técnicas de integração em larga escala (LSI).

A memória de computador deve armazenar dezenas de milhares de bits, arranjados em grupo de bits chamado "palavras", que determinam a dimensão básica da informação transferida, sejam memórias construídas com semicondutores (chips) ou com núcleos magnéticos.

Uma memória é "arranjada" (projetada) para suportar milhares de "palavras", onde cada "palavra" é armazenada em um endereço previamente selecionado. Os dados são escritos na memória, uma "palavra" por vez, e preservados para utilização futura, sendo lidos quando a informação é necessária.

Durante o intervalo entre uma escrita e uma leitura outras "palavras" podem ser armazenadas e lidas em outras localizações da memória. Os tamanhos típicos de "palavra" para minicomputadores são 12, 16 e 32 bits; para microcomputadores são 8 e 16 bits.

Normalmente, os outros registros de um computador são do mesmo tamanho da palavra de memória.

A capacidade típica de uma memória de microcomputador é 2K de "palavra", onde **K** é a abreviatura de **"KILO"** ou 1000, $K = 2^{10}$ ou 1024. Este valor de **K** é usado porque os projetos de memórias são orientados para tamanhos constantes de potências de 2.

Na literatura usual é comum o termo **KB**, KILO BYTE, onde o **BYTE** é uma palavra de 8 bits. Assim sendo, um computador com "palavra" de 16 bits pode ser citado como tendo 2 **BYTES** de "palavra".

Há uma outra categoria de memória, memória periférica de massa, fundamentalmente magnética (fitas, tambores, discos etc . . .) que apresenta um baixíssimo custo por bit, embora seja muitíssimo mais lenta que as memórias anteriormente consideradas.

Características Gerais

No que se refere à localização dos dados na memória estas dividem-se fundamentalmente em três tipos:

- **Memórias de Acesso Aleatório:** *A localização da informação é precisa e seu acesso é direto; por exemplo, a memória da CPU, seja com chips ou núcleos magnéticos.*
- **Memórias de Acesso Seqüencial:** *O acesso à informação é seqüencial, ou seja, necessita-se passar por uma série de outras informações armazenadas; por exemplo, as fitas magnéticas.*
- **Memórias Associativas:** *A informação é encontrada por analogia com o conteúdo do armazenado.*

O desempenho da memória na transição dos dados é indicado através de alguns tempos característicos, que apresentamos a seguir:

- **Tempo de Leitura:** *Indica o tempo transcorrido entre a localização e a disponibilidade da informação na saída da memória.*
- **Tempo de Escrita:** *Tempo transcorrido entre o endereçamento e a gravação efetiva da informação.*
- **Tempo de Acesso:** *Média entre os tempos de escrita e de leitura.*
- **Tempo de Ciclo:** *Soma dos tempos de acesso e de recuperação (necessário após uma operação de leitura ou de escrita).*
- **Freqüência de Transferência da Informação:** *Taxa com que a memória pode transferir os dados, na leitura ou na escrita, medida em bits por segundo.*

Para comparação, a tabela 1 apresenta os tempos de acesso de alguns tipos de memória.

TIPOS DE MEMÓRIA		TEMPOS DE ACESSO
discos e tambor magnético		5 a 100 μs
núcleos magnéticos		300 a 1000 ns
memórias semicondutoras	MOS	100 a 700ns
	BIPOLARES	10 a 200 ns

NOTAS:

microsegundo = $\mu s = 10^{-6} S$

nanosegundo = $ns = 10^{-9} S$

As memórias podem também ser divididas em voláteis e não voláteis:

- **Memórias Voláteis:** *As informações armazenadas são perdidas quando se deixa de fornecer potência ao circuito, por exemplo, as memórias semicondutoras de modo geral (chips).*
- **Memórias Não Voláteis:** *As informações armazenadas são mantidas mesmo quando falta alimentação ao circuito, por exemplo, as memórias magnéticas.*

Endereçamento X Informação de Memória

A memória central de um computador pode ser constituída de semicondutores (ROM, RAM, PROM, EPROM etc.) ou núcleos magnéticos. Em qualquer caso as informações armazenadas possuem endereços bem determinados e, portanto, as "palavras" que vão introduzir ou receber informações da memória devem ser constituídas de duas partes: uma contém a informação propriamente dita e a outra contém o código que permita o seu endereçamento.

Por outro lado, as unidades de processamento não podem dispor ou enviar as informações diretamente para a memória, necessitando de registros intermediários que armazenam o código de endereçamento e a informação.

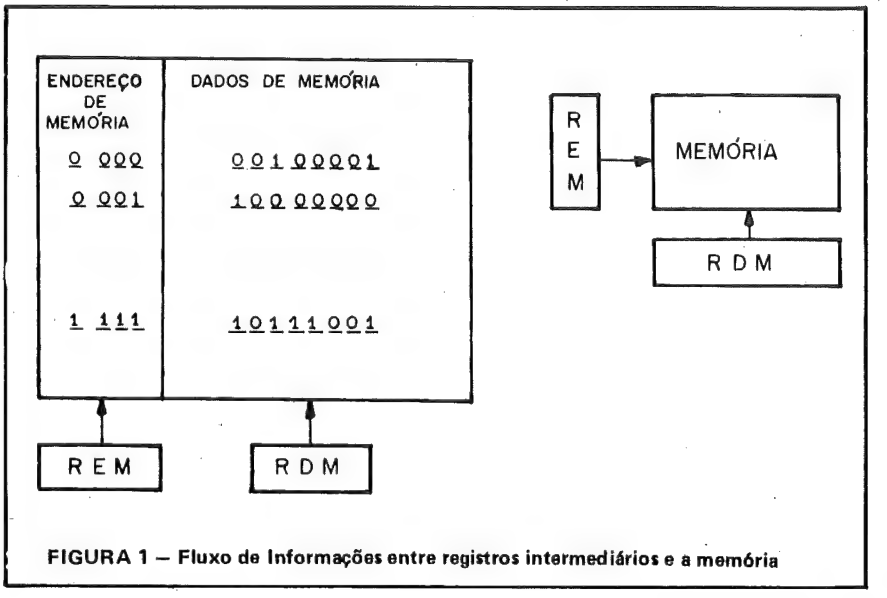


FIGURA 1 – Fluxo de Informações entre registros intermediários e a memória

O registro que armazena o endereço é conhecido como REGISTRO DE ENDEREÇO DA MEMÓRIA (REM) e o que armazena a informação como REGISTRO DE DADOS DE MEMÓRIA (RDM).

Suponhamos que uma memória possua 16 palavras de 8 bits, ou seja, sua capacidade é de 16 x 8 bits de memória. Isto significa que esta memória possui 16 endereços distintos, um para cada palavra de 8 bits cada uma, de modo que o registro de endereço de memória (REM) permita a formação de 16 códigos e para isso contenha 4 bits. Como a palavra de memória possui 8 bits, o registro de dados de memória (RDM) deve possuir 8 bits. A figura 1 apresenta o fluxo de informações entre os registros intermediários e a memória, bem como a estrutura da palavra com o código de endereço.

Para que um dado seja escrito na memória torna-se necessário, então, colocar o código de endereço no registro de endereço e o dado ou informação no registro de dados; um comando de modo de operação **ESCRITA/LEITURA** é então colocado em situação de **ESCRITA,** de modo que a informação é transferida para o endereço desejado.

Para lermos uma informação armazenada em um endereço de memória novamente o código de endereço é colocado no registro de endereço e o modo de operação é colocado em situação de **LEITURA,** o que faz com que a informação contida no endereço estabelecido seja transferida para o registro de dados.

Se um microcomputador possui 48 KB de memória, quantos bits são necessários para o REM? Quantos bits são necessários para o RDM e quantos bits de dados estão contidos no interior da memória?

Solução:

A especificação 48 KB significa que o microcomputador tem a capacidade de memória de 48 KB de palavras de 8 bits cada.

Por esse motivo a sua memória principal contém 48K = 48K x 1024 = 49152 palavras ou BYTES. O REM deverá ter capacidade para endereçar de 0 a 49151 posições de memória, necessitando portanto de 16 bits.

Caso o leitor não saiba como calcular este valor damos a seguir uma fórmula que deverá ser usada:

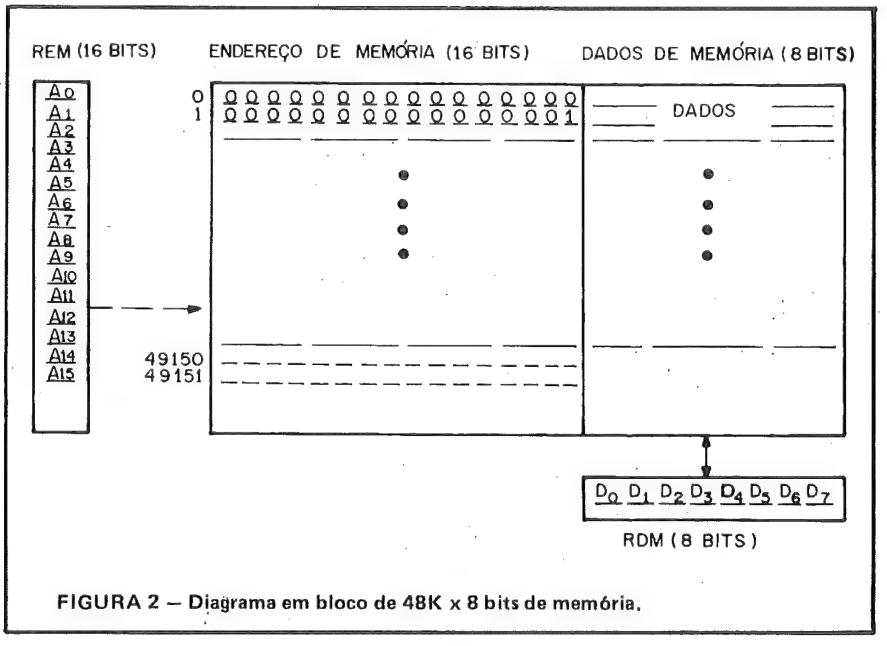


FIGURA 2 — Diagrama em bloco de 48K x 8 bits de memória.

$2^K \geq n$

Onde:
K = número de bits do REM
n = número máximo de palavras que a memória contém
Assim sendo, $2^K \geq 49151$; logo, K é igual a 16 bits.

Como a "palavra" de memória é 8 bits, o RDM terá 8 bits para dados, vide figura 2.

No próximo número apresentaremos as memórias de núcleo magnético.

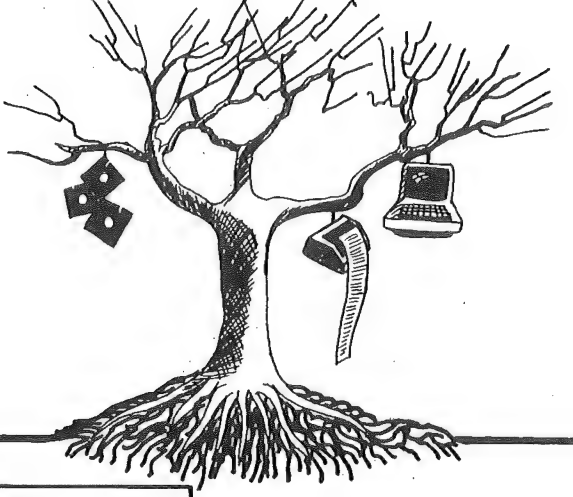
COMPLETE A SUA COLEÇÃO

INTERFACE PREÇOS ESPECIAIS:

1 EXEMPLAR — 550,00 —
2 EXEMPLARES — 1.050,00 —
3 EXEMPLARES — 1.500,00 —
4 EXEMPLARES — 2.000,00 —
5 EXEMPLARES — 2.500,00 —

ENVIE-NOS SEU NOME ENDEREÇO E UM CHEQUE PAGÁVEL NO RIO DE JANEIRO A FAVOR DE PRODIGT — PROC. TEC. E COMUNICAÇÃO LTDA EST. DO TINDIBA, 2380 - CEP 22.700 JACAREPAGUÁ — RJ —

produtos novos



μC 200

Em lançamento o **μC 200** da Scopus, que é um microcomputador desenvolvido para aplicações gerais de processamento, entrada de dados e teleprocessamento. Por suas características, pode ser utilizado por empresas de qualquer porte, pois seus recursos permitem a adequação do equipamento às necessidades de cada usuário.

O CPU do **μC 200** utiliza microprocessador INTEL 8080 ou 8085, 16 Kbytes de memória permanente (EPROM) e 64 Kbytes de RAM.

O teclado é composto de 126 teclas que operam por saturação de núcleo e o vídeo de 12" é em fósforo verde.

Este equipamento admite até duas unidades de disquete flexível de 8" de face simples ou dupla, atingindo a capacidade de 2,2 Mbytes. Também admite até duas unidades de mini disquete de 5 1/4", atingindo a capacidade de 256 Kbytes.

O **μC 200** pode ser conectado a qualquer impressora que possua interface no padrão RS-232C.

Opera com SOS (Sistema Operacional Scopus), que constitui-se numa evolução do CP/M expandido e aperfeiçoado, sendo compatível com o CP/M em versão original.

μC200

NOVO MICRO DA PROLÓGICA

A Prológica – Microcomputadores apresenta o microcomputador **CP-300**, o mais novo lançamento da empresa, que já está disponível no mercado. O equipamento, mais barato que uma televisão em cores, é intermediário do CP-500, o microcomputador pessoal mais vendido no Brasil, e do CP-200.

O **CP-300** é um equipamento modular que permite um futuro crescimento com a utilização de diversos periféricos e acessórios também fabricados pela Prológica, como a unidade de disco flexível, a impressora P-500 (que será lançada em junho) etc.

O equipamento, com uma memória de 64 KB (48 KB de RAM e 16 KB de ROM) é acoplado a um aparelho de televisão comum. Mas, a grande vantagem do **CP-300** é ser compatível com o software aplicativo do CP-500, em português, e com o TRS 80, o micropessoal norte-americano mais vendido no mundo. O **CP-300** opera em Basic e a Prológica coloca à disposição do usuário os seguintes aplicativos: contabilidade, contas a pagar e a receber, folha de pagamento e controle de estoque.

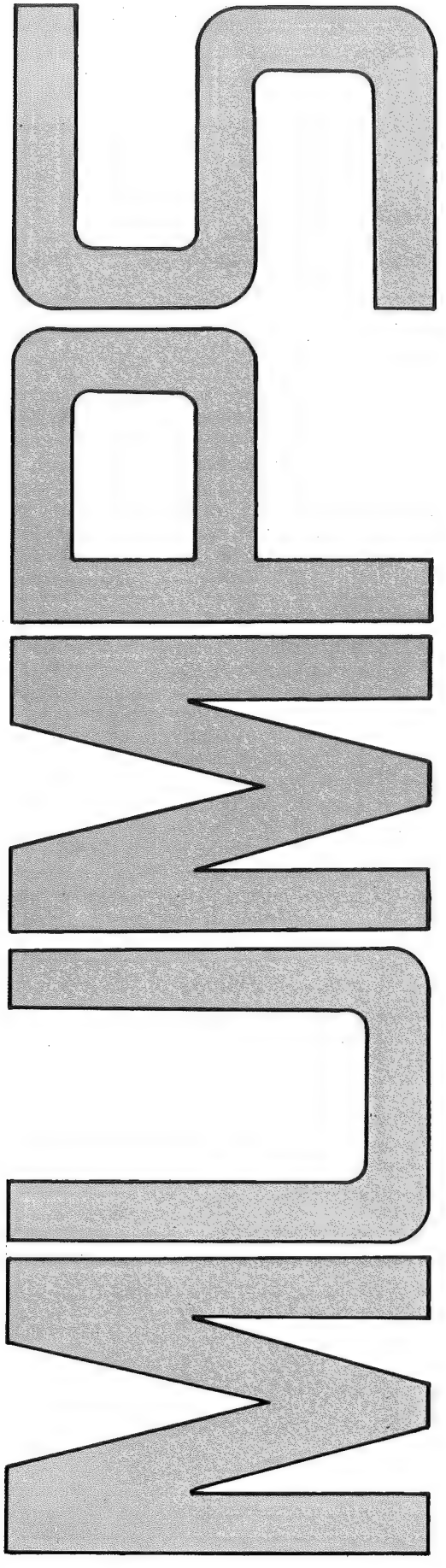
Com o lançamento do **CP-300**, cuja produção inicial prevista é de 1500 a 2000 unidades/mês, a Prológica se tornou o único fabricante a oferecer três diferentes tipos de microcomputadores pessoais.

OPÇÃO DE LAZER: ODYSSEY

Já está sendo comercializado o video-game da Philips, o **Odyssey**. Com produção anual prevista de 50 mil consoles e 200 mil cartuchos, o **Odyssey** custa Cr$ 150 mil, incluindo o telado alfa numérico, dois joysticks e um cartucho com três jogos. Com o teclado o jogador poderá modificar alguns parâmetros do jogo contido no cartucho, propondo inclusive novas alternativas em certos jogos.

Os cartuchos estão sendo lançados ao preço médio de Cr$ 18 mil, num total de mais de 20 jogos diferentes.

CP 300



CARACTERÍSTICAS E COMPARAÇÕES COM OUTROS SISTEMAS DE PROGRAMAÇÃO

1.ª Parte

Por T. Munnecke, R. F. Walters, J. Bowie, C. B. Lazarus e D. A. Bridger.

Esta é uma tradução do artigo escrito originalmente em inglês "MUMPS: Characteristics and Comparisons with other Programming Systems", publicado no Medical Informatics, **volume 2, nº 3, pág. 173-196, 1977. Foi traduzido pela MEDIDATA, que ao fazer esta tradução teve por objetivo oferecer à comunidade de usuários de computadores do Brasil mais um elemento para conhecimento e avaliação do Sistema MUMPS. Assim sendo, as opiniões contidas no documento são as dos autores acima citados.**

Os objetivos de projeto e as características do MUMPS (Massachusetts General Hospital Utility Multi-Programming System) estão descritos a seguir. O MUMPS é comparado com outros sistemas de programação, incluindo Basic e APL. As características e a utilidade de tipos diferentes de sistemas de banco de dados são comparadas com os recursos oferecidos pelo MUMPS.

O MUMPS foi projetado como uma linguagem interpretativa, voltada para o manuseio de cadeias de caracteres com recursos para o gerenciamento de banco de dados apropriados para os problemas de sistemas de informação médica. O sucesso deste projeto levou a muitas implementações diferentes, cada uma com seu próprio dialeto de linguagem. Em 1972, o **National Center for Health Services Research** e o **National Bureau of Standards** patrocinaram a criação do "Mumps Development Committee" para estudar a possibilidade de se criar uma nova padronização MUMPS, que fosse aceita pelos usuários e fabricantes em comum. Após anos de esforço, tal padronização foi adotada e as especificações foram enviadas para o **American National Standards Institute** para serem aprovadas.

Após a adoção da padronização, um grande número de fabricantes e grupos individuais implementou a nova versão em máquinas que variam de tamanho e vão de microcomputadores a grandes computadores. Também crescem os esforços para aumentar o número de máquinas que se utilizam dos interpretadores MUMPS pelo mundo inteiro, com centros importantes de interesse nos Estados Unidos, Japão e Europa. O compromisso por parte dos fabricantes de **hardware** diz bem da importância do papel que esta linguagem deve ter. As firmas de **software** também já consideraram a linguagem como um meio muito eficiente para o desenvolvimento e manutenção de pacotes comerciais para aplicações médicas. Só recentemente é que se deu valor a essa linguagem em áreas não-médicas, mas os usuários de áreas não-médicas já comprovaram uma economia de 75% do custo do desenvolvimento de aplicações comparáveis, usando hardware e técnicas comerciais (COBOL) convencionais.

A LINGUAGEM MUMPS FOI PROJETADA PARA SER FÁCIL DE SE APRENDER.

Tendo em vista estes benefícios e a relativa ignorância sobre o potencial da linguagem, preparou-se este relatório para fornecer às comunidades de processamento de dados e de assistência médica as informações sobre o MUMPS, e auxiliar na determinação de sua aplicabilidade para problemas específicos. As semelhanças e diferenças do MUMPS, em relação a outros sistemas de programação conhecidos, são discutidas aqui com detalhes a fim de que o leitor possa tomar conhecimento de seus pontos fortes e suas limitações. Os autores estão muito gratos pela contribuição de vários membros da comunidade MUMPS pela revisão deste manuscrito antes do seu envio para publicação.

O MUMPS é uma linguagem de programação interpretativa de alto nível e um sistema de gerenciamento de banco de dados. Ele é especialmente adequado para aplicações interativas que exigem um grande banco de dados dinâmico e compartilhado, e a manipulação eficiente de dados em forma de texto.

O desenvolvimento do MUMPS começou no **Laboratory of Computer Science** do **Massachusetts General Hospital** em Boston Massachusetts, em 1966. Uma pesquisa anterior sobre sistemas de informações médicas encontrou frustração e insatisfação com o estado da tecnologia de então. As experiências durante 1960 a 1965 com os sistemas de linguagem **assembler** não foram muito bem sucedidas devido ao enorme tempo de desenvolvimento e tempos de resposta excessivos, mesmo para modificações de programas mais simples. Por essas razões o pessoal do **Laboratory of Computer Science** decidiu fazer o projeto de um sistema eficiente de **time-sharing** (tempo compartilhado) para a gerência de dados clínicos. As características desse sistema foram padronizadas a exemplo do **JOSS**, uma linguagem interpretativa de alto nível desenvolvida pela **Rand Corporation** em 1964. A experiência também foi tirada dos descendentes do **JOSS**, tais como **TELCOMP** e **STRING-COMP** desenvolvidas por Bolt, Beranek e Newman, e **FILECOMP** especificado pela General Electric para o sistema MEDINET.

O objetivo do sistema MUMPS era combinar uma linguagem de alto nível simples, mas eficaz, com um sistema de manipulação de banco de dados fácil de usar. A linguagem MUMPS foi projetada para ser fácil de se aprender, com métodos simples de criação de programa, modificação e depuração. Os recursos da linguagem para manipular as cadeias de caracteres de tamanho variável e E/S em muitos terminais também eram exigidos. Um sistema de banco de dados hierárquico foi desenvolvido como parte integrante do MUMPS. Determinou-se que a estrutura hierárquica seria o método mais apropriado de manipular o complexo de dados demográficos, diagnóstico, resultados de laboratórios e outros dados necessários para a gerência de dados clínicos.

Um gerente de banco de dados foi projetado para facilitar o acesso à estrutura hierárquica da linguagem MUMPS. Recursos básicos foram desenvolvidos para implementar funções simbólicas de atualização e de recuperação. O projeto enfatizava o suporte de um banco de dados dinâmico sujeito a freqüentes alterações alternadas com consultas **on-line**. O programador MUMPS poderia levremente projetar tanto o conteúdo quanto a estrutura de dados que melhor se ajustasse à sua aplicação. Finalmente, o sistema foi implementado com um executivo compacto e de tempo compartilhado (time-sharing) para tornar eficiente o uso de todos os recursos num ambiente de minicomputadores.

A adequação de todos esses objetivos de projeto está hoje totalmente comprovada. O sistema MUMPS encontrou um lugar significativo tanto no meio médico quanto no não médico. Desde a implementação original no **Laboratory of Computer Science**, pelo menos sete outros dialetos e variações de MUMPS foram desenvolvidos. A fim de juntar essas opiniões divergentes e promover intercâmbio de programas, o **US Department of Health, Education and Welfare** e o **National Bureau of Standards** iniciaram o desenvolvimento da padronização do MUMPS. Esta especificação de padrões junto com outros documentos MUMPS para ensino e tradução da linguagem estão agora disponíveis no "MUMPS Development Committee".

A disponibilidade do MUMPS padrão ampliará a divulgação do MUMPS e facilitará a transportabilidade de muitas aplicações que já foram desenvolvidas.

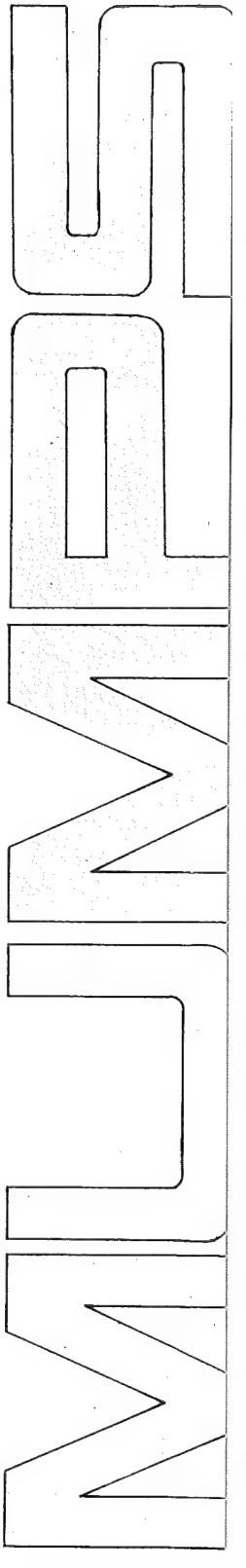
A fim de comparar o MUMPS com outros sistemas de programação, é necessário decifrar e comparar a sintaxe e semântica encontrada em manuais de linguagens e desvendar suas capacidades funcionais. Estas características podem então ser comparadas em base de igualdade. Entretanto, a combinação singular da linguagem com o sistema de banco de dados do MUMPS torna tal comparação muito difícil. Este relatório discutirá esses dois aspectos do sistema separadamente. Primeiramente, a linguagem padrão MUMPS será comparada com outras linguagens atualmente usadas nas aplicações de processamento de informações. Em seguida, as características dos sistemas atuais de gerenciamento de banco de dados serão descritas e seus recursos serão comparados àqueles oferecidos pelo sistema MUMPS.

O MUMPS é mais do que uma linguagem de programação. Ele é um sistema completo que combina em uma única sintaxe o que outros sistemas operacionais poderiam chamar de:

- uma linguagem de programação de aplicações;
- uma linguagem de controle de tarefas (JLC);
- um **Linkage Editor;**
- um gerente de banco de dados;
- um monitor de comunicação de dados.

Devemos ter em mente este **approach** do projeto da linguagem quando quisermos combinar as comparações das linguagens discutidas no final deste relatório.

As vantagens das linguagens de alto nível para o projeto de sistemas de aplicações já são conhecidas. As linguagens **assembler** não serão incluídas



O SISTEMA MUMPS ENCONTROU UM LUGAR SIGNIFICATIVO TANTO NO MEIO MÉDICO QUANTO NO NÃO MÉDICO.

nas discussões que se seguem. É suficiente dizer que os objetivos de desenvolvimento rápido e de facilidade de manutenção de programas são melhor atingidos por uma linguagem de alto nível, cujas estruturas e recursos sejam bem ajustadas ao problema de processamento da informação que se tenha em mãos.

INTERPRETADORES VERSUS COMPILADORES

A primeira característica a ser considerada é a natureza interpretativa do MUMPS. O MUMPS compartilha essa característica de interpretação, ao invés do método compilado mais comum, com várias outras linguagens, incluindo BASIC e APL. Com as linguagens interpretativas, os programas são executados diretamente de sua forma fonte ou simbólica. Não é necessário compilar ou traduzir um programa para uma outra forma antes que este possa ser executado. Este fator, por si só, pode representar uma economia de tempo significativa para o programador. Trabalhar somente na forma simbólica também facilita o desenvolvimento e manutenção do programa. Durante a depuração, os erros podem ser analisados e corrigidos rapidamente nos pequenos segmentos do programa, muitas vezes em uma única sessão interativa no terminal do programador. O esforço dispendido na manutenção de programas é bem menor em um ambiente interpretativo. Talvez o fator mais importante a se considerar e que favorece a interpretação é a flexibilidade aumentada do programador. Esta flexibilidade permite uma participação maior por parte dos programadores no atendimento às exigências do usuário final. As alterações ou acréscimos de especificações podem ser feitos rapidamente em um sistema de aplicações sem causar impacto às atividades de processamento dos sistemas em situação de produção regular.

Considerando os custos decrescentes de **hardware**, os custos crescentes de pessoal e a expansão da utilização de equipamentos de pequeno porte, cuja limitação principal não é mais o recurso CPU, os benefícios dos sistemas interpretadores em relação àqueles que requerem compiladores para gerar código mais eficiente tomam cada vez um maior vulto.

Um outro fator importante do interpretador foi que neste ambiente a necessidade de memória é grandemente diminuída. A sintaxe concisa do MUMPS pode gerar uma economia de memória na razão de dez para um com relação a um sistema equivalente codificado em linguagem **assembler**. Enquanto a natureza crítica do recurso memória tem diminuído de alguma forma nesses últimos anos pela redução dos custos, os sistemas operacionais ainda podem tirar vantagem de exigências de memórias reduzidas, de várias maneiras, sendo que uma das mais importantes é a possibilidade de reduzir o acesso a disco pela retenção de mais código na memória principal.

As reduções de espaço devem, naturalmente, ser confrontadas com a possível redução do tempo de execução dos interpretadores.

O **overhead** da interpretação pode se tornar significativo quando se lida com processos **CPU-Bound**, tais como a inversão de matrizes etc. Os processos de manipulação de informações, por outro lado, são raramente limitados pelo processador central. Eles estão mais limitados pela produtividade do manipulador de banco de dados ou pela velocidade de entrada e saída do terminal, já que em programas onde existe uma parte principal de diálogo interativo o processo geralmente está esperando pela entrada do usuário.

Um outro fator que atenua a diferença do tempo de execução entre código interpretado e compilado é a manipulação de cadeias de caracteres. O tratamento das operações de cadeias de caracteres é muitas vezes conseguida de forma similar, através de sub-rotinas. A maioria das funções comuns tais como a concatenação, a obtenção de sub-cadeias, a validação de sintaxes etc, são executadas por sub-rotinas "enlatadas", ativadas durante a interpretação de certas funções e operadores. Nas aplicações, onde a manipulação de cadeias representa uma atividade importante, a possibilidade de degradação provocada pela interpretação fica bem reduzida.

IMPLEMENTAÇÃO DO MUMPS

O MUMPS está atualmente implementado para uma grande variedade de tipos e marcas de máquinas. A implementação mais comum, entretanto, está nos minicomputadores e utiliza uma estrutura de multiprogramação. Neste projeto, a memória principal está alocada com base em partições dedicadas. Na inicialização, cada tarefa é designada para uma área específica, ou partição, na memória. Esta partição permanece ligada à tarefa até o seu término. As partições podem ser do mesmo tamanho ou de tamanhos diferentes, mas em quase todos os casos elas são fixadas na inicialização da tarefa. As tarefas são sempre residentes em memória, isto é, nunca ocorre **Swapp-out** ou **Check-point** durante sua execução. Assim, o número de tarefas ativas é limitado, explicitamente, pelo tamanho da memória principal. A decisão de se manter todos os usuários residentes na memória é baseada principalmente na necessidade de um tempo de resposta rápido. O **Overhead** de tempo associado à paginação é freqüentemente bem grande. Quando um sistema é usado em modo altamente interativo obtém-se vital eficiência pela residência na memória. Somente o programa em execução no momento (ou segmento do programa) é residente na partição neste exato momento. À medida que cada segmento é solicitado o interpretador lê o programa a partir do dispositivo de memória auxiliar do sistema e o sobrepõe ao segmento anteriormente ativo. A área de código e a área de dados em memória são mantidas em regiões isoladas que variam dinamicamente de tamanho em direções opostas. Este método permite a conciliação entre as exigências (conflitantes) de memória entre código e dados, isto é, o programador pode projetar

um programa extenso e uma tabela de símbolos reduzida, ou um programa pequeno com um grande número de variáveis. Os tamanhos da partição do MUMPS estão na faixa de 3 a 5 Kbytes.

O MUMPS é uma linguagem de procedimento simples organizada em linhas de código constituídas de um ou mais comandos básicos. Os nomes dos comandos são seguidos, opcionalmente, pelos argumentos que complementam a especificação da sua ação. Estes argumentos podem ser nomes de variáveis simples, como no caso de **KILL**, ou então expressões muito complexas como em **IF** ou **HANG**. As expressões em MUMPS podem consistir de combinações de variáveis, operadores e funções intrínsecas do MUMPS. As variáveis, como são usadas aqui, incluem tanto as variáveis locais, aquelas conhecidas apenas por cada tarefa particular ativa e as variáveis globais, ou comuns, que podem ser partilhadas por usuários diferentes e podem ser preservadas pelo sistema, depois que uma sessão interativa tiver terminado. Na maioria dos contextos, os dois tipos podem ser usados indistintamente. As variáveis locais podem ser criadas e destruídas livremente pelo programador; não há declarações explícitas de variáveis no MUMPS. O comando **SET** atribui valores às variáveis que são automaticamente criadas, se não existirem. O comando **KILL** elimina as variáveis especificadas da tabela de símbolos do usuário.

O MUMPS permite escalares e matrizes multidimensionais. As matrizes são armazenadas de forma esparsa, isto é, o espaço é somente utilizado pelos elementos aos quais são dados os valores. Este espaço é alocado dinamicamente pelo sistema operacional MUMPS. O padrão MUMPS tem somente um tipo de dado: as cadeias de caracteres de comprimento variável. Os valores das cadeias podem ser interpretados como inteiros, números reais, valores lógicos, ou cadeias, em função da sua utilização. Os operadores, então, não ficam restringidos na sua aplicação, já que eles podem "forçar" que interpretações apropriadas sejam feitas. A avaliação das expressões se procede estritamente da esquerda para a direita, havendo a possibilidade do uso de parênteses para indicar agrupamentos.

Muito do sucesso de MUMPS pode ser atribuído à escolha cuidadosa de operadores e funções. Além das operações aritméticas booleanas tradicionais, recursos são incluídos para a extração de cadeias, de concatenação, comparação e validação de sintaxe. As cadeias podem ser montadas, desmontadas, modificadas ou pesquisadas, com a facilidade que a maioria das linguagens de alto nível oferecem somente às expressões aritméticas. Esta atenção especial à manipulação de cadeias se deve ao compromisso do MUMPS com as aplicações de manuseio de informações.

A Entrada-Saída (E/S) no MUMPS é projetada para facilitar a programação interativa. Todos os dispositivos de E/S aparecem como arquivos seqüenciais que podem ser abertos por cada tarefa individualmente. O sistema de arquivos organiza esta função de retenção e coloca a tarefa no estado de "espera", se seu pedido não puder ser satisfeito. Os comandos **READ** e **WRITE** controlam o fluxo de informações de e para esses arquivos. Uma variedade de opções de formatação está disponível, incluindo especificações simbólicas para: nova linha, nova página, tabulação para colunas e alinha-

O MUMPS PERMITE ESCALARES E MATRIZES MULTIDIMENSIONAIS COM MÉTODOS SIMPLES DE CRIAÇÃO DE PROGRAMA.

mento de campo. Uma característica singular do comando **READ** do MUMPS é a capacidade de especificar um limite de tempo para a interrupção forçada de uma entrada. O uso deste **time-out** permite que um programa verifique o estado de um conjunto de terminais e automaticamente desvie para uma outra seqüência do programa quando um terminal estiver inativo, ou que forneça orientação quanto a resposta a ser dada, quando esta não for dada no tempo especificado.

Com o MUMPS, todo esforço tem sido feito para fornecer uma estrutura de E/S flexível, permitindo testes e avaliações rápidas do formato de saída e seleção de dispositivo (terminal).

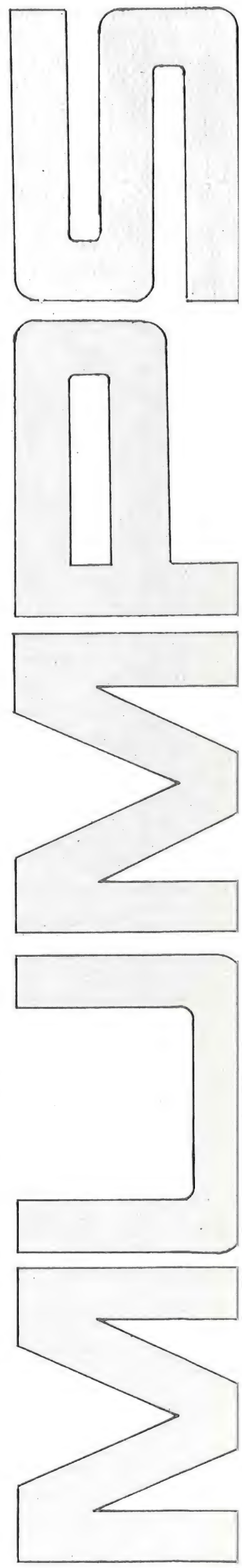
Talvez a característica mais singular do MUMPS seja a sua incorporação de um banco de dados hierárquico comum. O MUMPS foi especialmente projetado para manipular banco de dados textuais em uma forma lógica e estruturada. Um arquivo MUMPS ou matriz global e uma estrutura de dados hierárquica que consiste de camadas ou níveis. Cada nível é composto de um número de pontos de dados ou nodos. Estes nodos podem ter informações (cadeias) e/ou ligações com outros níveis (mais baixos).

As ligações circulares ou ascendentes em geral não são permitidas. Sintaticamente, um nodo específico é referenciado simbolicamente como um elemento em uma matriz N-dimensional: Λ AP (I, J, K). Aqui, o caractere (Λ) é usado para significar uma referência de arquivo global, AP (Arquivo de Pacientes) é o nome da matriz global e (I, J, K) é a seqüência de subscritos que especifica o nodo desejado. Cada posição na lista de subscritos representa um nível diferente da estrutura. A seqüência específica de subscritos pode ser considerada como sendo aquela que descreve um caminho através da estrutura e coloca informações em qualquer dos seus nodos. Tanto o conteúdo quanto a estrutura da matriz global podem ser alterados durante sua utilização. A ligação entre os níveis é executada automaticamente pelo sistema de arquivo.

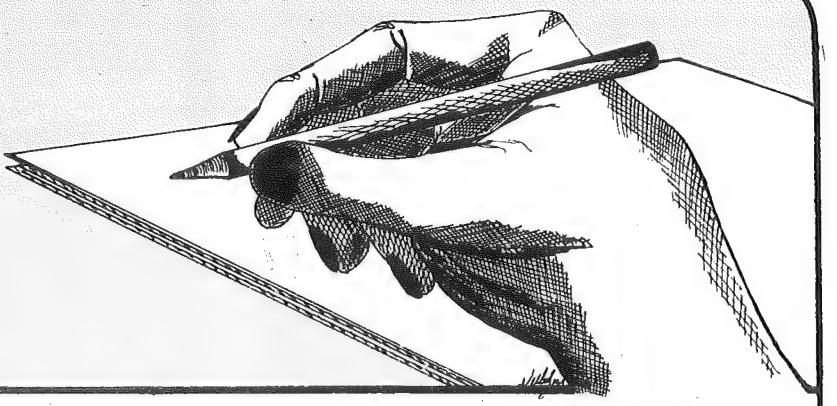
Esse método da linguagem MUMPS de referência aos arquivos de acesso randômico deve ser comparado ao método mais comum de leitura e gravação de elementos de dados ou registros.

Colocando-se as chaves de dados dentro da estrutura de variáveis nominais, consegue-se remover uma grande parte da confusão e trabalho ao se lidar com as áreas de **buffer** e compactação e descompactação de campos. As variáveis globais são buscadas e colocadas nos contextos específicos de variáveis. A linguagem MUMPS contém, adicionalmente, um comando específico (**LOCK**) para facilitar a sincronização dos processos que acessam simultaneamente o banco de dados e a proteção ao mesmo.

No próximo numero apresentaremos as comparações entre MUMPS, APL e BASIC.



cartas



NÃO É APENAS MAIS UMA REVISTA . . .

Quando um amigo me disse que estava colecionando uma nova revista de informática não dei muita importância, pois este mesmo colega coleciona várias revistas, e da mesma forma que aumenta vertiginosamente o desenvolvimento e aplicações de microcomputadores nos últimos anos, cresce também o número de publicações especializadas, de forma que se fosse inteirar-me de todas estaria lidando com tal quantidade de informações que acabaria por não entender nada.

Tive, porém, minha opinião modificada quando este colega mostrou-me um exemplar de sua revista, mais precisamente o nº 4. Percebi que INTERFACE preenchia plenamente os requisitos para os quais procurava respostas em diversas publicações, sem no entanto consegui-las a contento, tais como descrição de microprocessadores, desenvolvimento de sistemas, tanto em hardware quanto em software, e diversos outros, tão bem abordados nesta publicação, ressaltando a qualidade técnica dos textos e o zelo na apresentação dos mesmos.

Wanderley da Silva Claudio
Belo Horizonte — MG

Todos os seus elogios foram muito bem recebidos. Agradecemos e esperamos continuar atendendo a você e a todos os nossos leitores.

OBSERVAÇÕES IMPORTANTES

Somente 10 exemplares da revista INTERFACE nº 5 chegaram a este município, em 11-05-83. Este número mostra-se insuficiente para atender aos cursos técnicos desta cidade, bem como ao curso de Engenharia Elétrica da UFPB-Campus II. Sugiro, assim, a V. Sªs que tomem providências no sentido de, pelo menos, duplicar essa quantia a fim de que o numeroso contingente de alunos, os quais possuem um enorme interesse na área de Informática, possam ter acesso a essa maravilhosa publicação técnico-informativa.

Desejando-lhes longevidade aliada a um efetivo aprimoramento de sua publicação especializada, subscrevo-me.

Celso Emanuel Alves Oliveira
Campina Grande — PB

Suas observações são realmente importantes para nós. Já estamos providenciando distribuição mais efetiva em sua cidade.

APELOS/AJUDAS

Inicialmente devo parabenizá-los pela excelente revista e desejar-lhes vida longa. Tenho um TRS-80 color computer e desejaria me corresponder com possuidores deste computador para intercâmbio de idéias e experiências. Desde já fico muito grato pela publicação deste **apelo.**

Francisco Soares
Belo Horizonte — MG

Estou enviando junto a esta o pedido de renovação de assinatura.

Estou gostando muito desta revista pois ela tem sido usada como uma boa ferramenta de apoio para a Faculdade e também para ampliar os conhecimentos na área de informática.

Em alguns nºs passados foi abordado um assunto sobre a linguagem de programação MUMPS. Gostaria que me indicassem algumas pessoas que conheçam a linguagem MUMPS para procurar uma ajuda. Tenho que resolver um problema e gostaria de poder ter **ajuda** *de alguém.*

Se interessar a vocês posso mandar alguns programas em MUMPS pois já foi pedido por um outro assinante para ver como esta linguagem é construída.

Marco Aurélio de Souza Cabral
Rio de Janeiro — RJ

REALMENTE UMA REVISTA DE COMPUTADORES

Sou leitor assíduo da revista INTERFACE e acho que realmente é a única revista dentre as outras da área a qual se pode chamar realmente de uma revista de computadores. Embora seja nova, nota-se um grande esforço do editorial em fazer com que a revista se faça conhecida de um modo dilacerante e sempre com o intuito de aperfeiçoar-se cada vez mais.

Lendo a revista VEJA encontrei um artigo deveras interessante sobre QUITERON, um novo componente eletrônico que possivelmente vai revolucionar o mercado de computadores e possivelmente, quem sabe, lançar os computadores de 4ª geração. Já que a INTERFACE enfoca em especial a parte de Hardware, acho importante que se publique algo mais detalhado sobre o assunto, algo realmente do caráter e da qualidade da INTERFACE, tendo a certeza de que o artigo seria "devorado" com muito entusiasmo pelos leitores.

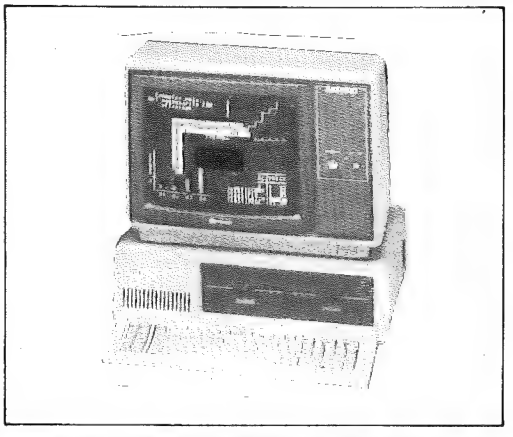
Marcelo Guidotti Palma da Fonseca
São Paulo — SP

Marcelo, seus elogios nos fizéram muito bem. Quanto à sua sugestão, foi devidamente anotada para uma futura abordagem.

JULHO DE 83

Interface

mercado de computadores



Como escolher um computador que seja adequado às suas necessidades? Hoje os preços de computadores já são acessíveis, e todos já podem possuir um, inclusive você.

Selecione aqui o seu microcomputador ou periférico e saiba onde adquiri-lo nos mercados do Rio, São Paulo, Belo Horizonte etc.

Características Técnicas & Preços

Microdigital TK 85. Venha dominá-lo.

Características Técnicas
- Linguagem BASIC
- 10 Kbytes de ROM.
- 16 ou 48 Kbytes de memória RAM.
- 40 teclas e 160 funções.
- Gravação de programas em fita cassete comum.
- Input e Output de dados.
- Vídeo: aparelhos de TV B&P ou colorido.
- Funções especiais HIGH-SPEED.
- Som Opcional.
- Joystick, impressora.

Preço de lançamento:
Cr$ 179.850,00 (16K)
Cr$ 249.850,00 (48K)
(Preço sujeito a alteração)

A primeira coisa que surpreende no TK 85 é o seu visual. Ele é compacto, leve e muito bonito. Se você esperar dele o desempenho de um pequeno computador, vai se surpreender outra vez: o TK 85 é um computador de grande capacidade e de grandes recursos. Acione o TK 85 e comece a dominá-lo. Você vai dominar, também, todas as situações. Resolver seus problemas domésticos ou profissionais. Vencer desafios e se divertir com jogos animados e inteligentes. Computador Pessoal TK 85. Uma fera às suas ordens.

MICRODIGITAL
Rua do Bosque, 1.234 - Barra Funda
CEP 01136 - Cx.P. 54.088 - São Paulo - SP
PABX 825-3355

REVENDEDORES: ARACAJÚ 224-1310 • **BELEM** 222-5122/226-0518 • **BELO HORIZONTE** 226-6336/225-3305/225-0644/201-7555 • **BLUMENAU** 22-1250 • **BRASÍLIA** 224-2777/225-4534/226-9201/226-4327/242-6344/242-5159 • **BRUSQUE** 55-0675 • **CAMPINAS** 32-3810/8-0822/32-4155/2-9930 • **CAMPO GRANDE** 383-6487/382-5332 • **CARUARU** 721-1273 • **CUIABÁ** 321-8119/321-7929 • **CURITIBA** 232-1750/224-6467/224-3422/243-1731/223-6944/233-8572/232-1196 • **DIVINÓPOLIS** 221-2942 • **FLORIANÓPOLIS** 23-1039 • **FORTALEZA** 226-4922/231-5249/231-0577/231-7013 • **FREDERICO WESTPHALEN** 344-1550 • **GOIÂNIA** 261-0333/224-0557 • **IJUÍ** 332-2740 • **ITAJUBÁ** 622-2088 • **LINS** 22-2428 • **LONDRINA** 22-4244/23-9674 • **MACEIÓ** 223-3979/221-6776 • **MANAUS** 237-1793 • **MOGI DAS CRUZES** 468-3779/208-6797 • **MURIARÉ** 721-1593 • **NATAL** 222-3212/231-1055 • **NITERÓI** 722-6791 • **NOVO HAMBURGO** 93-1922/93-3800 • **PELOTAS** 24-5139 • **PORTO ALEGRE** 26-8246/21-4189/24-1411/22-3151/24-0311/21-6109/24-7746 • **PRESIDENTE PRUDENTE** 22-2788 • **RECIFE** 241-4310/224-8777/224-3436/224-4327 • **RESENDE** 54-1664 • **RIBEIRÃO PRETO** 636-0586/634-4715/635-1195 • **RIO DE JANEIRO** 267-1093/252-2050/253-3395/264-0143/259-1516/232-5948/591-3297/222-6088/267-1339/329-4869/228-2650/246-4824/239-5612/542-3849/62-8737 • **SALVADOR** 248-6666/235-4184/247-5717 • **SANTA MARIA** 221-7120 • **SANTO ANDRÉ** 455-4962/444-7375/454-9283 • **SANTOS** 4-1220/32-7045/35-1792/33-2230 • **SÃO CARLOS** 71-9424 • **SÃO JOÃO DA BOA VISTA** 22-3336 • **SÃO JOSÉ DOS CAMPOS** 22-3968/22-7311/22-8925/21-3135 • **SÃO PAULO** 853-0164/853-0448/239-4122/36-6961/61-4049/881-1149/258-3954/212-9004/282-2105/212-3888/545-4769/227-3022/864-8200/222-1511/259-2600/282-6609/813-4555/814-3663/826-1499/521-3779/270-7442/210-7681/813-4031 • **SOROCABA** 32-9988 • **TAUBATÉ** 31-4137 • **UBERABA** 333-1091 • **UBERLÂNDIA** 234-8796 • **VIÇOSA** 891-1790/891-2258 • **MARÍLIA** 33-4109

...PONTO DECIMAL É PARA INGLÊS VER, ORA VÍRGULAS

O fato dos microcomputadores nacionais terem seus projetos baseados em microcomputadores estrangeiros, principalmente americanos e ingleses, trouxe alguns problemas para nós, usuários brasileiros. Um dos principais, como a linguagem de programação (Basic, geralmente), pode ser superado com um pouco de boa vontade.

Outros, porém, não têm solução assim tão fácil. É o caso dos acentos (que não existem) e do ponto (vírgula) decimal. Hoje resolveremos este último no Basic residente do microcomputador DGT-100 e para os que se utilizem do MBasic da MICROSOFT na versão para CP/M.

Diversas são as maneiras de se atingir este objetivo, mas nenhuma é tão eficaz e definitiva quanto a de efetuar o "aportuguesamento" no próprio interpretador Basic, pois assim estaremos eliminando o problema pela raiz.

Iniciaremos pela alteração do MBasic, por ser a mais fácil, procedendo da forma descrita abaixo.

O que estiver escrito em letra minúscula é a porção digitada pelo operador. Todas as respostas vindas do sistema estão em letras maiúsculas.

- Coloque o disco contendo o MBasic Drive B
- Coloque um disco que contenha o programa DDT no Drive A
- Inicialize o sistema
- Após o sinal de Pronto '> A' aparecer, digite:
  A > ddt b: mbasic. com < cr >
- O sistema responderá:
  DDT VERS. X.X
  NEXT PC
  6000 100
  –
- Digite agora:
  – s31d0 < cr >
  31D0 2C 2e < cr >
  – s31d5 < cr >
  31D5 2E 2c < cr >
  – s32df < cr >
  32DF 2E 2c < cr >
  – s354c < cr >
  354C 2E 2c < cr >
  – s3557 < cr >
  3557 2C 2e < cr >
- Neste ponto, a alteração já foi executada. Precisamos agora salvar o programa alterado em disco. Digite então:
  – ^c (control-c)
  A > save 95b: basicm . com < cr >
- Carregue agora o interpretador BASICM como de costume para o MBASIC. Efetue o seguinte comando:
  print using "# # # , # # # . # #";
  123456.78 < cr >
  123.456,78

Notou que o ponto foi, agora, para o seu devido lugar?

Para a modificação do DGT-100 se faz necessário o uso de um Programador de Eproms, o que torna, como já dissemos anteriormente, o processo um pouco mais complicado e não acessível a todos os usuários.

Devido ao fato de que não há padronização na operação dos programadores de Eproms existentes no mercado não entraremos em detalhes sobre como executar esta operação. Para os felizardos que têm acesso a um Programador, o processo é o seguinte:

- Desligue o DGT-100
- Após retirar os quatro parafusos inferiores que prendem a tampa e a fita de acabamento, localize a memória ROM U30 na placa impresso principal. Retire-a com muito cuidado do soquete (verifique que é uma EPROM 2716)
- Coloque-a no Programador de Eproms e transfira o conteúdo daquela para este
- Retire a Memória ROM do Programador e guarde-a com cuidado para evitar danos à mesma
- Proceda agora às seguintes alterações no conteúdo do programador:
  – Nos endereços 001AH, 025EH troque D3H para D1H
  – Nos endereços 001EH, 006CH, 0294H troque D1H para D3H
- Coloque uma Eprom apagada no programador e execute a gravação da mesma
- Coloque-a no lugar de U30 e ligue o DGT-100
- Faça o seguinte teste:
  Print using "# # # , # # # . # #";
  123456.78 < cr >
  Se tudo estiver ok o micro responderá:
  123.456,78
  READY
  –

Verifique que não há alteração alguma no que se refere à programação. Apenas a saída (quer seja no vídeo ou impressora) é que sofre a troca do ponto pela vírgula.

Alertamos para o fato de que se o seu DGT-100 ainda estiver na garantia qualquer alteração no mesmo poderá pô-la a perder. .

Qualquer dúvida referente a esta "Dica", por favor, escreva ou telefone para a Microsol.

Informamos também que temos para fornecimento a Memória ROM para o DGT-100, já alterada.

Microsol Ltda – Av. Pontes Vieira, 1855 – S 2/3
Fortaleza – CE – CEP 60.000
Telefone (085) 227-5878

MICROCOMPUTADORES

FABRICANTE	APLICAÇÃO	PREÇO CONFIGURAÇÃO MÍNIMA	PREÇO (*) SISTEMA: 2 DRIVES DE DISCO + IMPRESSORA + VÍDEO + CPU (RAM MÁXIMA)	MICROPROCESSADOR	MEMÓRIA RAM (MÁXIMA)
MICRODIGITAL TK 82-C	Pessoal	Cr$ 99.850,00	Cr$ 349.850,00 Gravador+64 KB+Printer	Z80 (8 bits)	64KB
TK85	PESSOAL	Cr$ 179.850,00	Cr$ 409.700,00 Gravador + 48 KB + Printer	Z80A (8 bits)	48 KB
PROLÓGICA CP-200	Pessoal	Cr$ 240.000,00	—	Z80 (8 bits)	16KB
CP-500	Pessoal	Cr$ 899.000,00	*Cr$ 3.229.400,00	Z80 (8 bits)	48KB
CP-700	Negócio	Cr$ 4.900.000,00	*Cr$ 6.000.000,00	Z80 (8 bits)	64KB
DIGITUS DGT-100	Pessoal	Cr$ 726.920,00	*Cr$ 3.270.000,00	Z80 (8 bits)	48KB
UNITRON AP II	Pessoal	Cr$ 1.098.380,00	*Cr$ 3.550.000,00	6502 (8 bits)	48KB
DISMAC D8000	Pessoal	Cr$ 598.000,00	———	Z80 (8 bits)	48KB
D8002	Negócio	—	* Cr$ 3.200.000,00	Z80 (8 bits)	48KB
ALFA 3000 SP	Negócio	—	*Cr$ 4.800.000,00	8080 (8 bits)	64KB
ALFA 2064	Negócio	—	* Cr$ 6.800.000,00	8080 (8 bits)	64KB
POLYMAX MAXXI	Pessoal	Cr$ 975.000,00	*Cr$ 3.000.000,00	6502 (8 bits)	48KB
HEWLETT-PACKARD HP-85A	Científica Pessoal	Cr$ 2.057.000,00	—	HP (8 bits)	16KB
SCHUMEC M100/85	Negócio Científica	Cr$ 2.154.515,00	*Cr$ 5.700.000,00	8085A (8 bits)	64KB
SPECTRUM MICROENCENHO	Pessoal Negócio	Cr$ 1.014.000,00	*Cr$ 3.600.000,00	6502 (8 bits)	48KB
COBRA COBRA 305	Negócio Científica	—	* Cr$ 5.900.000,00	Z80A (8 bits)	64KB
KEMITRON NAJA	Pessoal Negócio	Cr$ 828.850,00	*Cr$ 2.900.000,00	Z80A (8 bits)	48KB

RIO DE JANEIRO:

RV1 — CLAPPY COMPUTADORES E SISTEMAS LTDA.
Av. Rio Branco, 12 Lj/sl — Tels. (021) 253-3395/253-3170

RV2 — KRISTIAN ELETRÔNICA LTDA
Rua da Lapa, 120 Gr. 505 — Tel. (021) 252-9057

RV3 — IPANEMA MICRO
Rua Visconde de Pirajá, 540 Lj/106 — Tel. (021) 259-1516

RV4 — LOJA MICRO-KIT
Rua Visconde de Pirajá, 303 SI/210 — Tel. (021) 267-8291

RV5 — CASA GARSON/DIGIT-HALL
Rua Uruguaiana, 5 — Tel. (021) 252-2050

RV6 — TESBI — ENGENHARIA DE TELECOMUNICAÇÕES LTDA
Rua Guilhermina, 638 — Tels. (021) 591-3297/249-3166

RV7 — COBRA COMPUTADORES E SISTEMAS BRASILEIROS S.A.
Praia do Flamengo, 200 — 13º andar — Tel. (021) 265-7552

RV17 — MICROEQUIPO
Rua Alvaro Alvim, 33/37 — Gr. 517 — Tel. (021) 220-5059

RV18 — NASAJON SISTEMAS
Av. Rio Branco, 45 — Gr. 1311 — Tel. (021) 263-1241

SÃO PAULO:

RV8 — IMARÉS MICROCOMPUTADORES
Av. dos Imarés, 457 — Moema — Tel. (011) 61-0946

RV9 — COMPUTERLAND
Av. Angélica, 1996 — São Paulo — Tels. (011) 258-3954/258-1573/256-3307
Av. Barão de Itapura, 917 — Campinas — Tels. (0192) 31-9733/32-4155

Os preços constantes neste caderno são válidos até 15 de julho Foram obtidos através de médias a partir das informações dos próprios revendedores, assim sendo, pode ocorrer uma variação de 10% sobre o valor real.

Os revendedores ou fabricantes interessados em participar deste caderno deverão remeter sua lista de produtos com preços atualizados ou releasing de sua empresa para: REVISTA INTERFACE — MERCADO DE COMPUTADORES — Estrada do Tindiba, 2380 — Jacarepaguá — Rio de Janeiro — RJ.

VÍDEO	CARACTERES	SISTEMA OPERACIONAL	LINGUAGEM	SOFTWARE APLICATIVO	INTERFACE	REVENDEDOR (CÓDIGO)
TV	32 X 24	Monitor	Basic, Assembler	Jogos, gráficos finanças	—	RV1 RV2 RV3 RV4 RV5 RV6 RV8 RV9 RV10 RV12 RV15 RV16 RV19
TV	32 x 24	MONITOR	BASIC, ASSEMBLER	CONTABILIDADE FINANÇAS, JOGOS ORÇAMENTOS	—	RV1 RV2 RV3 RV4 RV5 RV6 RV8 RV9 RV10 RV12 RV15 RV16 RV19
TV	32 X 22	Monitor	Basic, Assembler	Jogos, gráficos finanças	—	RV2 RV3 RV4 RV5 RV8 RV9 RV10 RV12 RV15 RV16 RV19
Vídeo de 12"	64 X 16	DOS 500	Basic, Assembler	Orçamento, contabilidade, jogos	RS232C	RV1 RV2 RV5 RV8 RV9 RV10 RV12 RV15 RV16 RV19
Vídeo de 12"	80 X 24	DOS 700	Cobol, Basic, Fortran, Faturol C, Assembler	Faturamento, folha de pagamento, controle de estoque	RS232C	RV1,RV5,RV8,RV9, RV10,RV12,RV15 RV19
TV (Adaptada)	64 X 16	Monitor	Basic, Assembler	Orçamento, contabilidade, jogos	RS232C	RV1 RV2 RV3 RV4 RV5 RV6 RV8 RV9 RV10 RV12 RV15 RV16 RV18
TV	40 X 24	APPLE DOS	Basic, Mini Assembler	Orçamento contabilidade, jogos	RS232C	RV1 RV2 RV3 RV4 RV8 RV9 RV10 RV12 RV15 RV16 RV17 RV19
Vídeo de 12"	64 X 16	DOS	Basic, Assembler	Orçamento, contabilidade, jogos	RS232C S 100	RV1,RV5,RV8,RV9, RV10,RV12,RV18
Vídeo de 12"	64 X 16	CP/M (Placa Adicional)	Basic, Assembler	Orçamento, contabilidade, controle de estoque	RS232C S 100	RV1,RV5,RV8,RV9, RV10,RV12,RV18
Vídeo de 12"	64 X 16	DOS	Basic, Assembler	Orçamento, contabilidade, folha de pagamento, controle de estoque	RS232C	RV1,RV5,RV8,RV9, RV10,RV12
Vídeo de 12"	64 X 16	DOS	Basic, Fortran, Cobol, DDT, PIP, Assembler	Contabilidade, folha de pagamento, crediário	RS232C	RV1,RV5,RV8,RV9, RV10,RV12,RV18
TV	40 X 24	APPLE DOS	Basic, Polysoft, Assembler	Orçamento, contabilidade, jogos	RS232C	RV1 RV5 RV4 RV8 RV9 RV10 RV12 RV15 RV18
Vídeo de 5"	32 X 16	HP 8505	Basic, Assembler	Profissional (matemática, estatística)	RS232C	RV1,RV5,RV8,RV9, RV10,RV12,RV15
Vídeo de 12"	80 X 24	CP/M 2.2	Basic, Cobol, Pascal, Fortran	Processador de texto, controle de processo, contabilidade, finanças	RS232C S 100	RV1,RV5
TV	40 X 24	APPLE DOS	Basic, Mini Assembler	Orçamento, contabilidade, jogos	RS232C	RV1,RV3,RV5,RV8, RV9,RV10,RV12,RV15
Vídeo de 12"	80 X 25	SOM/E, Mumps	Assembler, Cobol, LTD, LPS, Mumps, Fortran IV	Processador de texto, faturamento, contabilidade, SPP, TANDEM	RS232C 3 terminais "on-line"	RV7,RV11,RV13
Vídeo de 12"	64 X 16	KDOS	Assembler, Basic	Orçamento, contabilidade, jogos	RS232C	RV14

RV10 — MICROSHOP
Av. Lorena, 652 — Tel. (011) 282-2105

RV11 — COBRA COMPUTADORES E SISTEMAS BRASILEIROS S.A.
Av. Angélica, 927 — Tel. (011) 826-4543

BELO HORIZONTE:

RV12 — COMPUCITY
Rua Tomé de Souza, 882 — Tel. (031) 226-6336

RV13 — COBRA COMPUTADORES E SISTEMAS BRASILEIROS S.A.
Rua Fernandes Tourinho, 500 — 1º/6º andar — Tel. (031) 225-4955

RV14 — KEMITRON LTDA
Av. do Contorno, 6048 — Tel. (031) 225-0644

PORTO ALEGRE:

RV15 — ADVANCING COMPUTER SHOP
Rua dos Andradas, 1560 — Conj. 518/5º andar — Tel. (0512) 26-8246

CURITIBA:

RV16 — COMPUSTORE
Rua Emiliano Perneta, 509 — Lj. 1/3 — Tel. (041) 232-1750/232-8814

BAHIA/PERNAMBUCO:

RV19 — OFICCINA MINI E MICROCOMPUTADORES
Av. Conselheiro Aguiar, 1555 — Lj. 4 — Boa Viagem — Recife — PE — Tel. (081) 326-9318/(071) 248-6666 R/268

PÓLO DE INFORMÁTICA

No período de 8 a 15 de maio, no Centro de Convenções de Olinda — Pernambuco, a comunidade de Informática e o público em geral tiveram a oportunidade de assistir e participar da II FEIRA NACIONAL DE INFORMÁTICA e do V CONGRESSO REGIONAL DE INFORMÁTICA, patrocinado pela SUCESU-PE, com o apoio do governo do estado de Pernambuco e o Banco do Nordeste.

SUCESU-PE MISSÃO CUMPRIDA

Com a realização dos eventos a SUCESU-PE — Sociedade dos Usuários de Computadores e Equipamentos Subsidiários de Pernambuco encerra a primeira parte de um trabalho que tem por objetivo tornar o Nordeste, além de um consumidor, também um gerador de tecnologia com a consolidação do PÓLO DE INFORMÁTICA DE PERNAMBUCO.

Durante oito dias cerca de 50.000 pessoas estiveram presentes no Centro de Convenções de Olinda, onde 50 expositores apresentaram seus produtos: micros, software, publicações técnicas (INTERFACE) e outros serviços.

COBRA — COMPUTADORES E SISTEMAS BRASILEIROS S/A E O TEOR DE SACAROSE

A COBRA esteve presente à II FEIRA NACIONAL DE INFORMÁTICA, com uma área de 120 m^2, mostrando em seu estande vários microcomputadores COBRA 305 e um minicomputador COBRA 520.

A COBRA exibiu uma mini agencia bancária equipada com o Sistema de Terminais Financeiros composto de Concentrador, Terminais Financeiros e de Consulta. Como novidade, a COBRA desenvolveu para seus computadores um sistema que permite efetuar o pagamento ao fornecedor de cana-de-açúcar pelo **teor de sacarose,** dentro da nova sistemática do Instituto do Açúcar e do Álcool. Numa segunda fase, ora em desenvolvimento, os computadores COBRA serão ligados diretamente às balanças e instrumentos de laboratório das Usinas, automatizando totalmente o processo.

ITAUTEC EXPÕE SEU VERSÁTIL MICROCOMPUTADOR I-7000

Resultado de projeto desenvolvido inteiramente no Brasil, o microcomputador Itautec I-7000 foi apresentado pela empresa no seu estande na II Feira de Informática.

Tendo como uma de suas principais características a "versatilidade", o I-7000 operou na Feira de Informática como processador de textos e como micro-sistema. Podendo ainda funcionar emulando terminais IBM da família 3270, ou em processamento distribuído, ou como terminal de entrada de dados.

O I-7000 possui o microprocessador NSC-800, memória RAM de 64 Kbytes expansível para 128 Kbytes, memória EPROM de 4 Kbytes, e opera com até 4 unidades de disquete com capacidade de armazenamento de 4 Mbytes.

O sistema operacional é o SIM/M, compatível com o CP/M, e as linguagens aceitas são Basic, Fortran, Cobol, Pascal e outras compatíveis com o CP/M. A impressora pode ser de 80 ou 100 cps, matricial, ou ainda do tipo daisy-wheel, para o processamento de textos. Existe teclado APL opcional, e saídas para gravador cassete e joystick.

MAXXI, O LANÇAMENTO DA POLIMAX NO NORDESTE

A POLIMAX — Sistemas e Periféricos S.A. apresentou no seu estande o seu mais recente lançamento, o microcomputador MAXXI, definido pela empresa como um "micro pessoal muito profissional".

Este equipamento é compatível com o **APPLE II PLUS**, o que significa aceitar mais de 5 mil programas aplicativos já disponíveis no mercado. Além da qualidade, pela incorporação da tecnologia POLYMAX, outro ponto destacado do MAXXI é a sua versatilidade e assegurada capacidade de expansão, através de periféricos e acessórios também disponíveis.

O MAXXI tem uma aplicação eficiente em pequenas empresas, entre profissionais liberais, e em áreas técnico-educacionais tais como universidades, escolas, centros de pesquisa e informação e centros de formação e treinamento de pessoal e executivos.

Em caráter particular e pessoal, na administração dos assuntos domésticos como controle de contas bancárias, elaboração do imposto de renda, controle do orçamento doméstico e ajuda nos estudos das crianças e jovens, com seus inúmeros programas educacionais.

INTERFACE SORTEOU O DGT-100 DO CONCURSO DO ASSINANTE

INTERFACE chegou, através da II FEIRA DE INFORMÁTICA, à Comunidade de Processamento de Dados do Nordeste.

Nosso estande foi um dos mais visitados, onde cerca de quinhentas pessoas tornaram-se assinantes de nossa revista, e onde recebemos os maiores elogios de técnicos locais e autoridades do mundo da Informática, que nos viram nascer no último INFORMÁTICA 82, no Rio de Janeiro.

Como havíamos anunciado na revista número 3, concurso do assinante, sorteamos o microcomputador DGT-100. Apesar das quinhentas assinaturas locais durante o evento, o assinante sorteado foi do Rio de Janeiro cujos dados damos a seguir:

Assinante: JOSÉ SOARES P. FILHO
Rua Padre Francisco Lanna, 59 - Apto 302 - Vila Isabel - RJ
Assinatura de 2 anos realizada em 21 de março de 1983.

FEIRA

SAIBA O QUE ACONTECIA PELO MUNDO QUANDO VOCÊ NASCEU

Essa é uma das atrações que a Scopus Tecnologia colocou à disposição do público em seu estande na II Feira Nacional de Informática, em Recife. Trata-se de um programa desenvolvido para rodar no microcomputador Scopus, contendo as principais notícias que foram manchetes dos jornais paulistas, mês a mês, de janeiro de 1920 a dezembro de 1970, além de resumos sobre História e Economia, por ciclos, nos mesmos 50 anos.

Qualquer visitante que quisesse saber o que aconteceu no mês do seu nascimento, apenas digitava o seu pedido em um dos teclados dos terminais Scopus para obter a resposta, em pouco tempo, através de uma impressora.

A configuração que a Scopus montou para o seu banco de notícias é um exemplo de gerenciamento de rede através do Microscopus. São três terminais de vídeo e três impressoras ligados a um concentrador MUX-108, que se conecta ao microcomputador, sendo que cada terminal pode acumular até 10 consultas.

Os destaques jornalísticos, que abrangeram pesquisas em 18 mil edições diárias dos seis principais jornais paulistas da época, são apresentados no texto original (ortografia, linguagem etc.). Essa pesquisa foi feita por nove estagiários de jornalismo, durante cinco meses, nos arquivos da Biblioteca Mário de Andrade, em São Paulo.

E os assuntos mais gerais, de História e Economia, foram desenvolvidos por especialistas nessas áreas, e cobrem as principais características dos movimentos mais importantes de 1920 a 1970.

Com o objetivo de implantar uma estrutura mais abrangente de vendas e atendimento aos clientes no Nordeste, a Scopus Tecnologia está abrindo uma filial em Fortaleza e ampliando seus serviços de assistência técnica na filial de Recife.

Atualmente a Scopus tem nove filiais em todo o Brasil, das quais duas estão no Nordeste — em Recife e Salvador. No Norte, a empresa atua através de uma **system-house** em Belém, no Pará.

Mas a Scopus está presente no Nordeste há vários anos com terminais de vídeo instalados nas universidades e empresas de processamento de dados. Também os microcomputadores — os Microscopus — já estão operando em várias localidades através do Banço do Nordeste do Brasil (BNB).

E justamente para melhor atender aos novos usuários dos seus equipamentos, a Scopus está montando uma estratégia para atingir outras praças do Nordeste que não estão cobertas pelas filiais.

LINHA DE PRODUTOS

A Scopus, empresa de capital 100% nacional, fundada em 1975, começou suas atividades fabricando terminais de vídeo. Com o investimento contínuo em novos projetos e no desenvolvimento de tecnologia própria, a Scopus ampliou sua linha de produtos com processadores, multiplexadores e microcomputadores.

Hoje, a Scopus lidera o mercado de terminais de vídeo com mais de 20 mil terminais operando em todo o País e mantendo uma posição de destaque no setor de comunicação e processamento de dados.

As várias famílias de terminais: TVA 1000, Sagitta, TVA 2000 e a nova linha de produtos Scopus compatíveis com a série 3270 da IBM são resultados comprovados de uma empresa que domina tecnologicamente o que produz e está sempre buscando soluções específicas para atender às necessidades do mercado brasileiro.

Outro resultado desse mesmo aprimoramento tecnológico, os microcomputadores da Scopus foram desenvolvidos para aplicações gerais de processamento de dados. Além de extensos recursos para facilitar o trabalho do operador e do programador, o Microscopus dispõe de vários sistemas de aplicação de uso geral e específico com soluções adequadas ao mercado brasileiro, como o sistema de gestão de vendas, pagamento de pessoal, cálculo e recálculo do FGTS etc.

A Scopus também oferece programas desenvolvidos exclusivamente para resolver problemas de entrada de dados, e que rodam no Microscopus.

balcão

Classificados do
Leitor:
Remeta seu anúncio
Para:
Revista INTERFACE
– Mercado de Computadores.
Estrada do Tindiba, 2380
Jacarepaguá
CEP 22700 – Rio de Janeiro – RJ

VENDO disquetes de 8 polegadas marca Memorex, por Cr$ 3.500,00 cada. Tratar com Sérgio Moura pelo telefone 249-4281.
Sergio Moura
Rua Honório, 475 apto. 104 F.
Rio de Janeiro – RJ

Vendo CP-500 com um disquete, modem 1200 bps e curso de Basic, por Cr$ 1.200.000,00. Tratar com Carlos, telefone 717-7863, após as 19 horas.
Carlos Augusto Pinheiro Alves
Rua Miguel de Frias, 48 apto. 804
Niterói – RJ

TK82-C, NEZ 8000. TROCO, COMPRO E VENDO SOFTWARE. Diversos jogos. Faço qualquer negócio.
André Koch
Av. Emancipação, nº 402 – sala 2
Tramandaí – RS

TROCO POR MICROCOMPUTADOR duas plantas de avião, originais dos EUA, uma de monoplano, de madeira com motor WV 1500 e 7,3 m de envergadura, preço base Cr$ 130.000,00; e a outra é de um biplano de madeira e alumínio, um ultraleve com motor de 125 CC de Kart e 5,30 m de envergadura, preço base Cr$60.000,00.
Paulo F. B. C.
Av. Ceará, 1890 apto. 32
Porto Alegre

Médico X Programas para TK (82, 85) – Vamos intercambiar?
Dr. Roberto de Araújo Santos
Rua Uruguay, 205 apto. 803
Rio de Janeiro – RJ

VENDO um microcomputador TK82-C com expansão de memória de 16K de RAM; livro de programação Basic (manual de TK); e programas Xadrez II e Torre, Senha, Palito da microsoft e livro de programação, com 45 programas prontos para rodar. Preço – Cr$ 130.000,00 (cento e trinta mil cruzeiros). Tratar com Adalber – Tel. 549-7774 – tarde ou fins de semana.
Adalber Kupcinskas Alencar
Rua Afonso Celso, 718 – apto. 23
São Paulo – SP

VENDO OU TROCO PROGRAMAS PARA OS MICROS: CP-200, TK82-C e XZ-81. Possuo o Simulador de Vôo, Nautilus, Pac-Man, Bowling e muitos outros.
Eduardo Medeiros – Tel. (016) 625-5796.
Rua Eliseu Guilherme, 1076
Ribeirão Preto – SP
Atendo também por reembolso postal

Estamos formando um clube para o fornecimento de programas e dicas de programação.
Luís Guilherme Damiani – Tel. (011) 831-6645
Rua Umburanas, 504
São Paulo

LIVROS PARA TK, NEZ, CP

45 PROGRAMAS

Prontos para Rodar

ARQUIVOS, ESTOQUE, PLANO CONTÁBIL, AGENDA TELEFÔNICA, INVASORES, CAÇA AO PATO, APAGUE A TRILHA, JOGO DA VELHA, FORCA, DADO, TABELAS, TABUADAS, CONVERSÃO DE COORDENADAS, MÉDIA, PROGRESSÃO, FIBONACCI, BIO RITMO, RENUMERADOR DE LINHAS EM CÓDIGO, ETC.., ETC..,

5 Edicao **CR$ 3.000,00**

APLICAÇÕES SÉRIAS

C/ Programas Listados por Impressora.

FOLHA DE PAGAMENTO, BALANCETE, CONTAS A RECEBER, A PAGAR, CORREÇÃO MONETÁRIA DAS CONTAS DO BALANÇO, CORREÇÃO DAS CONTRIBUIÇÕES DO IAPAS, CADASTRO DE CLIENTES, CONTA BANCÁRIA, TABELA PRICE, ESTATÍSTICA, CORREÇÃO DE PROVAS, EDITOR DE TEXTOS, RAM TOPER. SUBROTINAS EM CASSETTE, CHAINING PROGRAMAS, CONTANDO OS BYTES DAS LINHAS, DO PROGRAMA, DAS MATRIZES, ECONOMIZANDO MEMÓRIA, ETC... ETC...
INCLUINDO:
CONHECENDO A IMPRESSORA, VALE A PENA? VEJA AMOSTRA DO PAPEL.
PROJETO COMPLETO DE TECLADO MECÂNICO, COM LAY-OUT DOS CIRCUITOS IMPRESSOS, DOS PAINÉIS E GABINETE, ETC...

CR$ 3.000,00

FIRMWARE PARA TK, NE Z

Circuito Leitor de Eprom **CR$17.500,00**

Ligado entre a expansão de memória e o micro, lê programas de uma EPROM pré-programada, com o uso da função USR. É só ligar para usar os programas. Não requer alimentação externa ou alterações no micro.

Eprom's Pré-Programadas

Funções I	Cr$ 9.500,00
Ram Toper	Cr$ 9.500,00
Paredão (jogo)	Cr$ 9.500,00

SOFTWARE EM CASSETTE

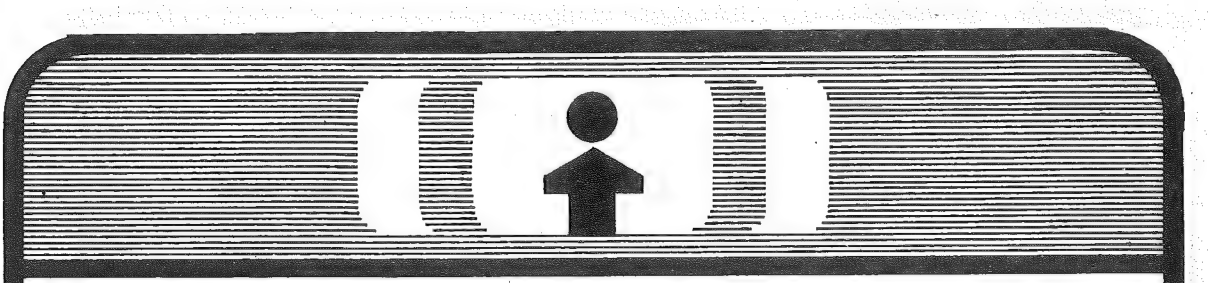
Solicite relação de programas.

À VENDA NAS LOJAS ESPECIALIZADAS
DESPACHAMOS PARA TODO O BRASIL MEDIANTE CHEQUE NOMINAL COM 10% PARA FRETE E EMBALAGEM.

MICRON
ELETRÔNICA COMÉRCIO E INDÚSTRIA LTDA.
Av. S. João 74 - Telefone 22-4194 S. José dos Campos Est. de São Paulo

Páginas de Serviço

* MINI E MICROCOMPUTADORES
* PERIFÉRICOS
* COMPONENTES
* SOFTHOUSE
* CONSULTORIA
* BUREAUX
* INSTALAÇÃO E MANUTENÇÃO



CBI

CENTRO
BRASILEIRO DE INFORMÁTICA

O Centro Brasileiro de Informática é uma entidade fundada por professores universitários e profissionais da área de Informática, que dentre outras atividades constituem a Consultoria Técnica da revista INTERFACE.

Os objetivos da CBI são:

- Criação, desenvolvimento e treinamento em tecnologia voltada para as reais necessidades brasileiras.
- Assistência e cooperação tecnológica com empresas, universidades e centros de pesquisa.
- Preparação e difusão de palestras, debates e cursos de interesse técnico na área de processamentò de dados, tanto a nível de software quanto a nível de hardware.

Para atingir estes objetivos a CBI conta com a sua colaboração. Participe e você terá toda a infra-estrutura para se desenvolver.

Cumprindo seus objetivos, dois importantes convênios foram assinados:

- CBI – APPD-RJ (Associação dos Profissionais de Processamento de Dados do Rio de Janeiro).
- CBI – DIGITUS (empresa fabricante do microcomputador DGT-100).

Responsáveis Técnicos:
Engº Cesar de Araújo Lima
Msc Manuel Lois Anido
Engº Luiz Tadeu B. Navarro
Msc Tarcisio Neves da Cunha
Engº Cesar da Costa
Engº Ivan da Costa
Eng. Ney Acyr de Oliveira
Msc. André Gil Rubens

CURSOS PARA JULHO/AGOSTO

MICROPROCESSADOR Z-80
(Revista Interface)

Arquitetura: hardware: funcionamento e projetos; software: conjunto de instruções e programas aplicativos; aulas práticas.

Eng. Ney Acyr da Oliveira
Msc. André Gil Rubens
Desenvolvimento de hardware e software de microcomputadores da EMBRATEL e consultores técnicos da revista INTERFACE.

MANUTENÇÃO DO MICROCOMPUTADOR PESSOAL DGT-100
(Em convênio com a DIGITUS)

Técnicas de manutenção, reparo, análise de defeitos.

Eng. Cesar da Costa
Eng. Ney Acyr de Oliveira
Msc. André Gil Rubens
Especialistas em hardware e manutenção de sistemas digitais.

PROGRAMAÇÃO MUMPS
(Em convênio com a APPD-RJ)

Técnicas de programação, soluções de casos práticos.

Eng. Ivan Costa
Engenheiro Eletrônico, Analista de sistema e computação

CP/M – SISTEMA OPERACIONAL
(Em convênio com a DIGITUS)

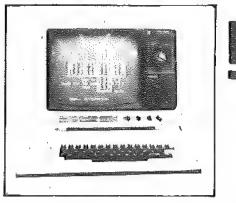
Conceitos básicos sobre as funções do sistema operacional CP/M e sua utilização em microcomputadores.

Msc. Manuel Lois Anido
Engenheiro Eletrônico, Mestre pela COPPE-UFRJ, projetista de microcomputadores.

Informações e inscrições:
CBI – Centro Brasileiro de Informática
Av. Passos, 115 – Sala 215
TEL.: 233-1123

INSCREVA-SE JÁ

INFORMÁTICA



INFORMÁTICA

A INFORMÁTICA é a indústria que mais cresce no mundo
a que paga os melhores salários
e a única carente de técnicos especializados

FORMAÇÃO DE TÉCNICOS ESPECIALIZADOS

PROGRAMAÇÃO DE MINI E MICROCOMPUTADORES
Objetivo

Ao final do curso o aluno estará capacitado a projetar e documentar programas em Assembler, Basic, Cobol, Pascal, Mumps e outras linguagens de programação em mini e microcomputadores, além de provê-lo de embasamento necessário para atuar como programador Júnior em CPD.

MANUTENÇÃO DE MINI E MICROCOMPUTADORES
Objetivo

Ao final do curso o aluno estará capacitado a elaborar rotinas de manutenção preventiva e corretiva, usar os principais equipamentos de testes digitais, além de capacitá-lo a projetar e documentar programas de testes para manutenção do hardware através do software de mini e micro-computadores.

PRÉ-REQUISITO - 2º grau
DURAÇÃO: 1.250 h - 12 meses
HORÁRIO: 2ª a 6ª das 19:00 às 22:00 hs - Sáb. das 8:00 às 11:20 hs

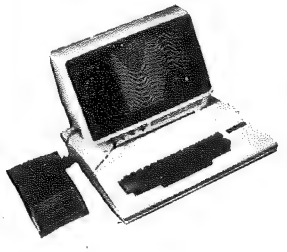
ESTÁGIO: 200 hs
Estágio operacional
Laboratório próprio
de Microcomputadores
e Eletrônica digital

CENTRO EDUCACIONAL CARVALHO DE MENDONÇA

ENGEMICRO
ENGENHARIA EDUCACIONAL DE MICROPROCESSAMENTO

Rua Evaristo da Veiga, 20 - Tel. 220-8820/220-7009

CERTIFICADO: REGISTRO NO SISTEMA OFICIAL DE ENSINO



MANUTENÇÃO AUTORIZADA

VENDAS DE PROGRAMAS

LANÇAMENTO

Excepcional programa de banco de dados para TK 82-C, Sinclair e NEZ 8000/16 K.
preço - Cr$ 15.990,00

Temos também suprimentos:
Formulários contínuos, disketes, fitas p/ impressoras.

Despachamos para todo o Brasil mediante Ordem de Pagamento ou Cheque nominal com acréscimo de 10% para frete e embalagem.

CURSO DE BASIC

Faça sua reserva cursos noturnos c/prática em computador e estágio garantido.

VENDA DE MICROCOMPUTADORES

CP 500 - D8000 - TK 82-C - D8002 - DGT 100.

TESBI Engenharia de Telecomunicações Ltda.
Demonstrações e Venda: Rua Guilhermina, 638 - RJ.
Tel.: (021) 591-3297 e 249-3166/Caixa Postal 63008

TESBI — Engenharia de Telecomunicações Ltda.

PROGRAMAS

(*) Banco de Dados - TB II	Cr$ 15.990,00
(**) Banco de Dados - TB I	Cr$ 9.990,00
(*) Cálculo de Lajes maciças	Cr$ 9.990,00
(**) Folha de Pagamento	Cr$ 15.990,00
(*) Xadrez II	Cr$ 6.890,00

(*) TK 82/Sinclair/NEZ 8000 (**) DGT 100/D8000/D8002

Anexo incluso cheque nº _____ do Banco _____ no valor de Cr$ _____

Meu nome: _____

Meu endereço: _____

CEP: _____

TRANSITRON

Comércio de Componentes

- TTL
- C.MOS
- ICL 7107
- 2114
- 2708
- 2716
- 2732
- LINHA Z80
- LINHA Z80A
- ELETROLÍTICO
- TRANSÍSTOR
- POLIESTER
- TÂNTALO
- PLATE
- RESISTOR
- FUSÍVEL
- SOQUETE
- CONECTOR C. IMP.
- APAGADOR DE EPROM

Cr$ 55.000,00

TRANSITRON ELETRÔNICA LTDA.
Rua dos Gusmões, 353, 3º andar, cj. 31
Fones 221-2959 - 221-2701 - 223-5187
Telex - 37982 Representante em B. H.
Rua Eng. Antonio Guerra, 174, cj. 401
Fone 332-9580 - Sr. Rogério.

LIVRARIA CIÊNCIA MODERNA LTDA.

LIVROS PARA SINCLAIR ZX-81 - CP 200 - TK-82C

• HURTLEY - The Sinclair ZX-81 Programming for Real Applications	8 159.00
• BAKER - Mastering Machine Code on Your ZX-81	10 619.00
• MAUNDER - The ZX-81 Companion Real Time Graphics Information Processing	8 159.00
• HARTNELL - 49 Explosives Games for the ZX-81	8 979.00
• CHARLTON - The Gateway Guide to the ZX-81 and ZX-80 More Than 70 Programs	8 159.00
• HARTNELL - Getting Acquainted with Your ZX 81 s/E More Than 80 Programs	8 159.00
• TOMS - The ZX-81 Pocket Book	8 979.00
• HARTNELL - Making the Most of Your ZX-81	8 979.00
• MELBOURNE - Not Only 80 Programs for the Sinclair ZX-81 - 1 K	9 730.00
• SANTOS LIMA - 45 Programs Prontos para Rodar em TK-82C e NE Z8000	3 000.00
• GROSJEAN - The Best of SYNC	8 159.00

PEDIDOS PARA:
AV. Rio Branco, 156
Lojas 217/230
Centro
20043 - RIO DE JANEIRO - RJ
TELS. 262-1989 e 262-2789

CEAPRO
TREINAMENTO E ASSESSORIA TÉCNICA

CURSOS DE ESPECIALIZAÇÃO PROFISSIONAL

MICROCOMPUTADORES
MICROPROCESSADORES

SOFTWARE
BASIC
ASSEMBLER

HARDWARE
INTERFACES DO 8080/85
MICROPROCESSADOR Z-80
MICROPROCESSADORES 8080/85
LÓGICA DIGITAL I e II
AMPLIFICADORES OPERACIONAIS

TELEPROCESSAMENTO
TELEPROCESSAMENTO I - HARDWARE
TELEPROCESSAMENTO II - SOFTWARE

BANANA-85
MICROCOMPUTADOR PARA DESENVOLVIMENTO DE SOFTWARE E HARDWARE
REVENDEDOR AUTORIZADO

AULAS PRÁTICAS COM
MICROCOMPÙTADORES NACIONAIS
KITS E LABORATÓRIOS DE ELETRÔNICA DIGITAL

TURMAS COM 20 ALUNOS
CURSOS FECHADOS PARA EMPRESAS

AV. PRESIDENTE VARGAS 590/GR. 217
RIO DE JANEIRO **Tel. (021) 233-5239**

LOJA MICRO-KIT
TUDO SOBRE MICROCOMPUTADOR

1º ANIVERSÁRIO PARABÉNS

Venha à nossa loja e fale com quem entende

CURSOS

Basic p/adultos e crianças, com método próprio comprovadamente eficiente; Professores c/mestrado em ENGENHARIA DE SISTEMAS; mais de 20 cursos aplicados. Turmas pequenas, aulas práticas com MICROCOMPUTADOR.

VENDA DE MICROCOMPUTADOR

Unitron AP II, Digitus, TK e CP 200.
Financiamento em até 24 meses.

PROGRAMAS

Comerciais e Jogos p/APPLE, Unitron, Polymax, Digitus TK e CP 200.

SUPRIMENTOS

Disquetes, Caixa p/Disquetes, Formulários Contínuos etc.

VENDA DE LIVROS E REVISTAS

Despachamos para todo o Brasil.

Rua Visconde de Pirajá, 303 S/Loja 210
Tels. (021) 267-8291 521-4638
CEP 22410 – Rio de Janeiro
Rua Visconde de Pirajá, 365 – Sobreloja 209
Ipanema

MC INFORMATICA

TUDO EM MICROCOMPUTADORES

- Cursos de programação com APOSTILA PRÓPRIA E AULAS PRÁTICAS em diversos MICROCOMPUTADORES
- Todas as principais marcas de MICROCOMPUTADORES pelo menor preço com crédito direto em até 24 MESES.
- Programas prontos ou por encomendas tanto de jogos quanto comerciais.

VENHA CONHECER HOJE
A ERA DO FUTURO

MICROCENTER INFORMÁTICA LTDA.

Rua Conde de Bonfim, 229 – Lojas 310 e 312 – Galeria Cinema III
Tel. 264-0143 – CEP 20520 – Tijuca
– Rio de Janeiro - RJ

LHM
COMÉRCIO E REPRESENTAÇÕES LTDA.

SOFTWARE DISPONÍVEL

- *Contabilidade Geral*
- *Contas a Pagar*
- *Mala Direta/Arquivos*
- *Editor de Textos*
- *Folha de Pagamento*
- *Administração de Imóveis*
- *Visiplot/Visitrend*
  e Vários Outros

APPLE/TRS-80/
UNITRON/POLYMAX

NAJA/DGT-100/
CP-500/MICRO-ENGENHO

L.H.M. — SOFTWARE-HOUSE
AV. FRANKLIN ROOSEVELT, 23
GRUPO 1203 — TELS.: 262-5437
CEP 20.021 — R.J.

OFERTAS Kristian

MICROCOMPUTADORES

DGT-100 Cr$ 220.000, x 3 — Grátis 18 JOGOS
CP-500 Cr$ 790.000, — Grátis 18 JOGOS
CP-200 Cr$ 100.000, x 2 — Grátis 18 JOGOS
TK82-C Cr$ 49.925, x 2 -- Grátis 10 JOGOS
ainda UNITRON Ap II, Mem 64K,
Impressoras, etc...
(Preços sujeitos a modificações)

PROGRAMAS PRONTOS EM FITAS

JOGOS TK ou CP
- VISITA AO CA$$INO
- MIDWAY
- PASSAGEM PARA O INFINITO
- 10 JOGOS EXCITANTES PARA 1K
- 2ª DIMENSÃO
- JORNADA NAS ESTRELAS E MUITO MAIS!

JOGOS: DGT-100 OU CP-500
- SCARFMAN
- PENETRATOR
- SUPER-NOVA
- VIAGEM A VALKYRIA
- ASILO 1
- AVENTURAS
- DEFENSE COMMAND
- E MUITO MAIS!

LEASING E CRÉDITO DIRETO!

LITERATURA
- MICRO-SISTEMAS
- INTERFACE
- JORNAL TK-CP
- IMPORTADOS

+ CURSOS DE BASIC GRÁTIS
NA COMPRA DE QUALQUER MICRO

DESPACHAMOS PARA TODO O BRASIL!

APLICATIVOS
- CONTROLE DE ESTOQUE
- CONTAS A PAGAR/RECEBER
- MALA DIRETA/CADASTRO
- FOLHA DE PAGAMENTO
- VÍDEO-CLUBES
- ESTATÍSTICOS
- SOFTWARE SOB ENCOMENDA

Kristian ELETRÔNICA LTDA.
Rua da Lapa, 120 Gr. 505
Rio de Janeiro - RJ
Tel.: (021) 252-9057

A CERTEZA DE UM BOM NEGÓCIO

FAIRCHILD — FAIRCHILD SEMICONDUTORES LTDA.
Transístores, Diodos de Sinal e Zeners. . .

GENERAL SEMICONDUCTOR INDUSTRIES INC.
Transístores, Diodos Transzorb. . .

IBRAPE — IBRAPE IND. BRAS. DE PRODUTOS ELETRÔNICOS E ELÉTRICOS LTDA.
Transístores, Diodos de Sinal e Zeners. . .

ICOTRON — ICOTRON S/A IND. DE COMPONENTES ELETRÔNICOS
Transístores, Capacitores de Poliéster Metalizado e Eletrolítico. . .

MOTOROLA SEMICONDUCTOR PRODUCTS INC.
Transístores, Circuitos Integrados, Retificadores, Tiristores. . .

SOLID STATE SCIENTIFIC INC.
Transístores, Circuitos Integrados. . .

TECCOR — TECCOR ELECTRONICS INC.
Tiristores, DIACS, SCR, TRIACS. . .

TELEDYNE SEMICONDCUTOR
Transístores, Diodos de Sinal e Zeners. . .

TEXAS INSTRUMENTS INC.
Transístores, Circuitos Integrados. . .

Kaprom

Teleimport Eletrônica Ltda.
Rua Sta. Ifigênia, 402, 8/10º andar - CEP 01207 - São Paulo
Fone: 222-2122 - Telex (011) 24888 TLIM-BR
(Solicite nosso catálogo geral de componentes)

RELAÇÃO DOS ANUNCIANTES

CÓD.	ANUNCIANTE	PÁGINA
01	Microdigital Computadores Pessoais	2ª Capa
02	Magnex	3ª Capa, 64
03	Prológica	4ª Capa
04	Clappy Computadores e Sistemas	3
05	Logus	7
06	Curso Internacional	7
07	Monk Micro Informática Ltda	10
08	Nasajon	10
09	Telesp	11
10	Polimax	21
11	Filcres	33
12	Advancing	30
13	BK	39
14	Microdigital Computadores Pessoais	52
15	Micron	58
16	CBI	60
17	Engemicro	61
18	Tesbi	61
19	LHM Comércio e Representações Ltda	62
20	Ceapro Treinamento e Assessoria Técnica	62
21	Microcenter	62
22	Microkit	62
23	Transitron	62
24	Ciência Moderna	62
25	Kristian Eletrônica Ltda	63
26	Teleimport Eletrônica Ltda	63
27	Litec	63

LITEC
Livraria Editora Técnica Ltda.

A maior livraria da América Latina especializada em INFORMÁTICA, COMPUTAÇÃO E ELETRÔNICA.

Mais de 3.000 títulos em português, espanhol e inglês em permanente exposição.

Rua dos Timbiras 257 - 01208 São Paulo Tel. (011) 220-8983 cx. postal 30.869

CARACTERÍSTICAS TÉCNICAS

MÓDULO BÁSICO

- Sistema dual de processadores: Z80 e 6502
- 64 KB de memória em RAM dinâmica (250NS)
- Monitor em EPROM de 2 KB
- Interpretador Basic residente em ROM de 10 KB
- Saída de vídeo à cores e preto/branco
- Vídeo de 24 Linhas por 40 Colunas (caracteres 5x7)
- Vídeo de 24 Linhas por 80 Colunas (caracteres 5x7)
- Saída de vídeo à cores padrão PAL-M
- Saída de vídeo à cores em RGB analógico
- Gráfico à cores em baixa resolução: 40h x 48v e 80h x 48v em 16 cores.
- Gráfico à cores em alta resolução: 280h x 192v em 8 cores
- 3 (três) conjuntos de caracteres selecionáveis por software
- Interface para gravador cassete de áudio
- Saída paralela para impressora
- Saída serial para modem e outros periféricos
- Teclado profissional separado com 93 chaves
- Teclado numérico completo
- 6 (seis) conectores para expansão
- Alimentado por fonte chaveada de alto rendimento
- Design moderno e funcional
- Compatível em hardware e software com APPLE II

OPCIONAIS PARA EXPANSÃO

- 128 KB de memória em RAM
- 2 (duas) unidades de disco flexível 5 1/4" (143 KB cada)
- 2 (duas) unidades de disco flexível de 8", formato IBM 3740 e também dupla face e dupla densidade capacidade 1 MB cada.

PERIFÉRICOS

- MG-PEN caneta de luz para interação direta com o vídeo
- JSTK-1 "Joystick" usado para controle de jogos
- TVX12 Televisor modificado com provisão para entrada direta do sinal de vídeo, para maior definição dos caracteres, fósforo branco, capacidade de 24 linhas por 80 colunas. Ainda mantém o funcionamento como televisor.
- TVC14 Televisor à cores de 14" com entrada RGB direta para melhor definição de caracteres e gráficos. Ainda mantém o funcionamento normal do aparelho.
- MV12 Monitor profissional de vídeo, tela de fósforo verde é indicado para aplicações comerciais.
- MGP - 100 Impressora serial 100 CPS, 132 colunas.

SOFTWARE

- Suporta SGD/MX - Sistema de Gerenciamento de Disco - Magnex
- Suporta o DOS 3.3
- Suporta o sistema operacional CP/M 2.2 e similares

APLICAÇÕES

- Na área administrativa, financeira e comercial, para cálculos estatísticos, quadros sinóticos, dados, etc.
- Na área Técnico-Científica, em pesquisas e desenvolvimento.
- Para profissionais liberais, como médicos, advogados ou outras profissões onde sejam necessários uma maior agilização e estocagem de informações.

Magnex Eletrônica Ltda.
Rua Dr. Thyrso Martins nº 100 - CEP 04120 - Vila Mariana - São Paulo - SP
Tel.: 570-2872 - 549-2232 Telex: (011) 4837 DLTD

GRUPO Daiya

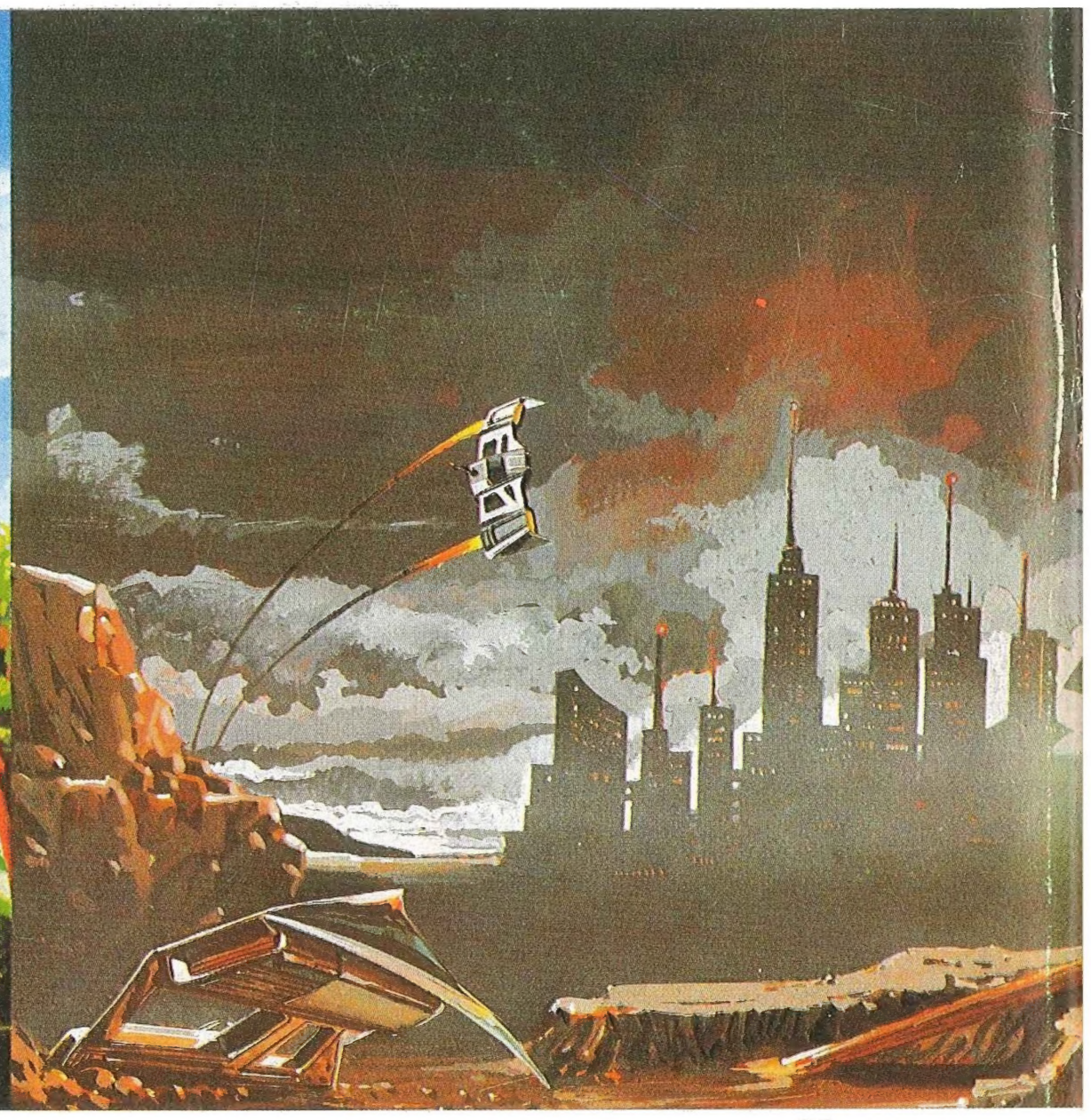
MAG

MG

8065 O Microcomputador que coloca o futuro ao seu alcance

MAGNEX

Para que futuro você está educando seu filho?



Os dois usam computador.

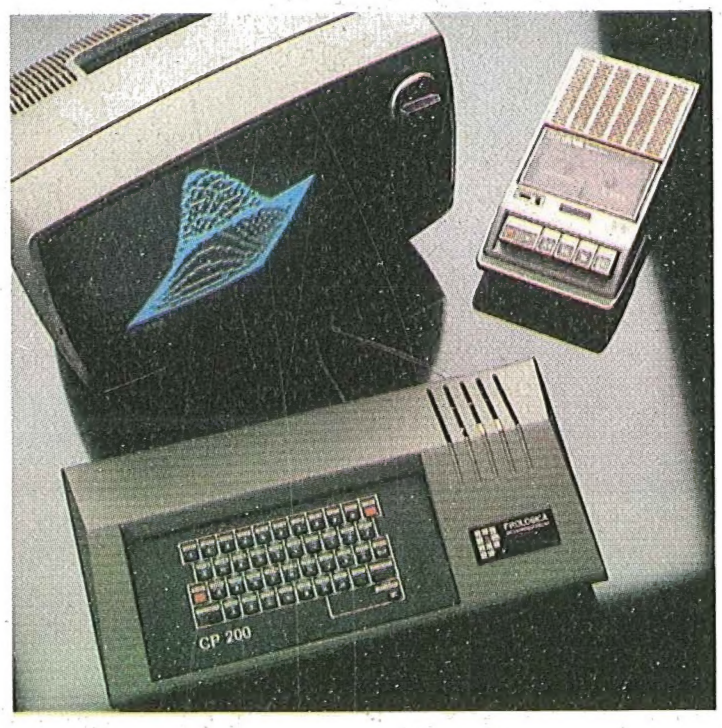
"Assim como toda educação emana de alguma imagem do futuro, toda educação emana alguma imagem do futuro." (Alvin Toffler)

O CP 200 *da Prológica é simples de operar, custa menos do que um tv a cores e faz importantes trabalhos de interesse de toda a família. Com ele você e seus filhos aprendem a linguagem "Basic" e ficam aptos a programar qualquer tipo de computador, participando e criando o momento atual que já é chamado de "a era da informática".*

Basta ligar o CP 200 *a um televisor e a um gravador para você ter um computador completo em sua casa.*

Assim como o extrato de tomate, o liquidificador, o durex, o automóvel, a máquina de escrever e a calculadora, o CP 200 *vai simplificar sua vida.*

E vai dar mais tempo para você e sua família criarem um futuro melhor.



Veja *o que você faz com o* CP 200:

- *Aprendizado em linguagem* Basic
- *Divertidos jogos e passatempos eletrônicos*
- *Orçamento doméstico*
- *Controle de conta bancária*
- *Aulas de matemática e física*
- *Gráficos e cálculos científicos*

SOLICITE DEMONSTRAÇÃO NOS PRINCIPAIS MAGAZINES.

PROLOGICA
microcomputadores

Av. Eng.º Luiz Carlos Berrini, 1168 - SP

AL - Maceió - 221-4851 - **AM** - Manaus - 234-1045 - **BA** - Salvador - 247-8951 - 235-4184 - **CE** - Fortaleza - 226-0871 - 231-1295 - 226-4922 - DF - Brasília - 226-1523 - 273-2128 - 225-4534 - 226-4327 - 242-6344 - **ES** - Vitória - 229-1387 - 222-5811 - **GO** - Goiânia - 224-7098 - 225-8598 - 224-4657 - **MA** - São Luís - 222-6696 - **MT** - Cuiabá - 321-2307 - **MS** - Campo Grande - 383-1277 - Dourados - 421-1052 - **MG** - Belo Horizonte - 201-7555 - 226-6336 - 225-3305 - 222-3196 - 227-0881 - Betim - 531-3806 - Cel. Fabriciano - 841-3400 - Juiz de Fora - 212-9075 - Uberlândia - 235-1099 - 235-6600 - Viçosa - 891-2445 - **PA** - Belém - 228-0011 - **PB** - João Pessoa - 221-8232 - 221-6743 - **PR** - Curitiba - 224-5616 - 243-1731 - 224-3422 - 223-2323 - 232-2793 - Ponta Grossa - 24-0057 - **PE** - Recife - 221-0142 - 221-5774 - **PI** - Teresina - 222-0186 - **RJ** - Campos - 22-3714 - Rio de Janeiro - 264-5797 - 221-5141 - 240-1099 - 266-4499 - 253-3395 - 252-2050 - **RN** - Natal - 222-3212 - **RS** - Caxias do Sul - 221-3516 - Gravataí - 88-1023 - Novo Hamburgo - 93-1922 - Porto Alegre - 26-8246 - 42-0908 - 27-2255 - 21-4189 - Sta. Maria - 221-7120 - **RO** - Porto Velho - 221-2656 - **SP** - Araçatuba - 23-8021 - Assis - 22-1797 - 22-2200 - Barretos - 22-6411 - Campinas - 2-4483 - 32-4145 -Jundiaí - 434-0222 - Marília - 33-5099 - Mogi das Cruzes - 469-6640 - 468-3779 - Mogi Guaçu - 61-0256 - Piracicaba - 33-1470 - Presidente Prudente - 22-3165 - Ribeirão Preto - 625-5924 - 625-5926 - 635-1195 - São Joaquim da Barra - 728-2472 - São José dos Campos - 23-3752 - 22-7311 - São José do Rio Preto - 32-2842 - Santos - 33-2230 - Sorocaba - 33-7794 - **SC** - Blumenau - 22-6277 - Campos Novos - 44-0196 - Criciúma - 33-1436 - Florianópolis - 22-9622 - 22-6757 - Itajaí - 44-1524 - Joinvile - 33-7520 - Rio do Sul - 22-0557 - **SE** - Aracajú - 224-1310

P P R